NPS ARCHIVE
1998.12
TSIRIDIS, K.

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

TIME DOMAIN SIMULATION OF MFSK COMMUNICATIONS SYSTEM PERFORMANCE IN THE PRESENCE OF WIDEBAND NOISE AND CO-CHANNEL INTERFERENCE

by

Konstantinos Tsiridis

December 1998

Thesis Advisor: Jovan Lebaric
Co-Advisors: Clark Robertson
David Jenn

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1998	Master's Thesis

4. TITLE AND SUBTITLE
TIME DOMAIN SIMULATION OF MFSK COMMUNICATIONS SYSTEM PERFORMANCE IN THE PRESENCE OF WIDEBAND NOISE AND CO-CHANNEL INTERFERENCE

5. FUNDING NUMBERS

6. AUTHOR(S)
Konstantinos, Tsiridis

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Research and Development Office

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(maximum 200 words)***

In this thesis models of MFSK digital communications systems were developed using Matlab Simulink and Communications Toolbox. The models were employed to verify MFSK performance in the presence of additive noise and predict MFSK performance for additive noise and co-channel interference. Results are presented for bit-error rate as functions of the signal-to-noise and signal-to-interference power ratios. The results for coherent detection of MFSK (M=2,4,8) in the presence of additive white Gaussian noise show excellent agreement with the theory. On the other hand, simulation results for the probability of bit error for non-coherent detection of MFSK differ (-21% average) from the theory suggesting a possible 'systematic' error in the Communications Toolbox implementation of the non-coherent MFSK detection.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Simulink, Communications Toolbox, MFSK coherent-non coherent detection.	140
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFI- CATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited

TIME DOMAIN SIMULATION OF MFSK COMMUNICATIONS SYSTEM PERFORMANCE IN THE PRESENCE OF WIDEBAND NOISE AND CO-CHANNEL INTERFERENCE

Kontantinos Tsiridis
Captain, Hellenic Army
B.S., Hellenic Military Academy, 1988

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING
AND
MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1998

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

ABSTRACT

In this thesis models of MFSK digital communications systems were developed using Matlab Simulink and Communications Toolbox. The models were employed to verify MFSK performance in the presence of additive noise and predict MFSK performance for additive noise and co-channel interference. Results are presented for bit-error rate as functions of the signal-to-noise and signal-to-interference power ratios. The results for coherent detection of MFSK (M=2,4,8) in the presence of additive white Gaussian noise show excellent agreement with the theory. On the other hand, simulation results for the probability of bit error for non-coherent detection of MFSK differ (-21% average) from the theory suggesting a possible 'systematic' error in the Communications Toolbox implementation of the non-coherent MFSK detection.

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

TABLE OF CONTENTS

I. INTRODUCTION....1

II. SIMULINK AND COMMUNICATIONS TOOLBOX....3

III. MODEL OF A MFSK PASSBAND DIGITAL COMMUNICATION AND ITS BASEBAND EQUIVALENT....19

A. BASEBAND EQUIVALENCE FOR PASSBAND SIGNALS AND SYSTEMS....19

B. MONTE CARLO SIMULATION....22

C. M-ARY FREQUENCY- SHIFT KEYING (MFSK)....25
1. Bit Error Probability for Coherent MFSK....26
2. Bit Error Probability for Non-Coherent MFSK....28

D. MFSK MODELS AND BLOCK ANALYSIS SIMULATION....29
1. Coherent MFSK Model (Noise Case)....29
2. Block Analysis....30
3. Coherent MFSK Model for MFSK with Interference and Additive Noise....39

IV. SIMULATION, ANALYSIS , AND PERFORMANCE VERIFICATION FOR WIDEBAND NOISE....45

A. PERFORMANCE VERIFICATION FOR WIDEBAND NOISE....45
1. Results for Coherent Binary FSK (BFSK)....46
2. Results for Coherent 4FSK....50
3. Results for Coherent FSK....55
4. Results for Noncoherent Binary FSK (BFSK)...60
5. Results for Noncoherent 4FSK....69
6. Results for Noncoherent 8FSK....74

B. OBSERVATIONS....79

V. SIMULATION AND PERFORMANCE ANALYSIS FOR CO-CHANNEL INTERFERENCE....83

A. INTERFERENCE IN DIGITAL COMMUNICATION SYSTEMS.....83

B. PERFORMANCE OF MFSK DIGITAL COMMUNICATION SYSTEMS IN THE PRESENCE OF ADDITIVE GAUSSIAN NOISE AND CO-CHANNEL INTERFERENCE.....84
1. Results For Coherent Binary FSK (BFSK).....85
2. Results For Coherent 4FSK.....89
3. Results For Coherent 8FSK.....92
4. Results For Noncoherent Binary FSK (BFSK)...96
5. Results For Noncoherent 4FSK.....99
6. Results For Noncoherent 8FSK.....102

C. OBSERVATIONS.....106

VI. SUMMARY AND CONCLUSIONS.....111

APPENDIX A.....113

APPENDIX B.....115

APPENDIX C.....117

APPENDIX D.....119

APPENDIX E.....123

LIST OF REFERENCES.....127

INITIAL DISTRIBUTION LIST.....129

ACKNOWLEDGMENT

Numerous individuals assisted me in completing this thesis. Many thanks to Dr. Jovan Lebaric for his technical support, clear explanation and supervising. I would also like to thank Prof. Clark Robertson and Prof. David Jenn for their advice and help.

Finally, I would like to thank my wife Antigoni who always supports me with love and encouragement, my parents and my country who gave me the means to acquire my knowledge.

I. INTRODUCTION

A. OBJECTIVES

In many military scenarios a large number of different communication systems operate simultaneously from the same site or platform, ideally with as little mutual interference as possible under the constraints of available frequency allocations and the physical separations of the antennas. It is important to predict the effects of co-channel interference such that measures may be taken to reduce the interference to a tolerable level.

The objective of the thesis is to model an M-ary frequency-shift keying (MFSK) communication system in the time domain using Simulink and MATLAB/SIMULINK Communications Toolbox. MFSK performance is verified for additive white Gaussian noise (AWGN). In addition, MFSK performance for simultaneous AWGN and co-channel interference is determined. This is accomplished by obtaining estimates via computer simulation of the bit error rates under these conditions.

B. METHOD

Using Simulink and the Communications Toolbox allows one to concentrate on system modeling rather than on the

details of code development in Matlab or some other computer language. The block-diagram based SIMULINK time domain modeling is a convenient tool for the visualization of communication signals at various stages of transmitters and receivers and for "Monte Carlo" type simulations of their performance. The results of Monte Carlo type computer simulation were used to establish the bit error rates, under realistic conditions of noise and interference and for different values of transmitter/receiver and channel parameters such as the power ratios of the desired and interfering signals.

In Chapter II the Simulink and Communications Toolbox is described. Chapter III discusses the models used for the simulation. The simulation results for coherent and noncoherent MFSK (M=2, M=4, M=8) in the presence of AWGN are verified in Chapter IV, and in Chapter V the results are discussed for the case involving the presence of both AWGN and interference. Finally, Chapter VI presents a summary and conclusions.

II. SIMULINK AND COMMUNICATIONS TOOLBOX

A. INTRODUCTION

Simulink is a program for modeling and simulating dynamic systems including linear, nonlinear, discrete-time, continuous-time, and hybrid systems. As an extension to Matlab, this environment adds many features specific to dynamic systems while retaining all of Matlab's general-purpose functionality. Model definition and model analysis are the two steps in a Simulink computer simulation. In practice, these two steps are often performed interactively as the model is created and modified in order to achieve the desired behavior.

Simulink represents systems as block diagrams. Using mouse driven commands, one can create models whose parameters can be edited or changed using the keyboard. Also, operations can be clicked-and-dragged, and results can be displayed in real time during a simulation. The models can be analyzed either by choosing options from the Simulink menus or by entering commands in Matlab's command window. The results, which can be viewed while the simulation is running, are made available in the Matlab workspace after a simulation has been completed. Since Simulink is built on top of the Matlab environment, it has many capabilities,

which can be further enhanced with application-specific toolboxes. Generaly speaking, Simulink combines the power and ease-of-use of an application package with the flexibility and extensibility of a language [Ref 2,1].

B. TIME DOMAIN SIMULATION

In order to evaluate the time domain simulation, the terms "modules" and "single-point format" need to be explained first. The modules of a simulation program can be regarded as subroutines that operate on the data, which is provided as input. The program calls the first module, which processes the data and returns the results. Then it calls the second module to process the data delivered from the first module and so on. This continues until all modules have been called, upon which the program repeats from the beginning with new input data. The term "single-point" refers to the amount of data that can be processed on each call of a module. When only one data point (only one time instant) can be processed or one data point is passed for computation from module to module, such simulation format is called "single-point data." On the other hand, if a whole block of N data points (N time instants) is processed on each call, the format is called "block data." This means that for an equal amount of total data, single-point format calculations are substantially more time consuming than the

block data operations since single point data calculations incur at least N times as much calling overhead as block data simulations for the same (N) number of time instants. The single-point format (which is the only alternative when a system includes feedback loops) processes only time-domain signals (periodic or aperiodic). Usually, single-point data is regarded as one sample from a long, aperiodic signal, and thus a single-point simulation will exhibit the same transient behavior as the system being simulated. Also, single-point simulations can be arbitrarily long in duration without special provisions. This makes the simulation more convenient when collecting data for performance statistics [Ref 6].

C. CONSTRUCTING A SIMPLE MODEL

Simulink uses block diagrams to represent dynamic systems. Defining a system is much like drawing a block diagram. The "building blocks" can be copied from various block libraries, which can be either built-in (Simulink or Communications Toolbox libraries) or user-created. The built-in Simulink is organized into sub-libraries according to block function or nature. In order to assemble a system, blocks can be copied from an existing library and connected using the mouse and new block parameter values inserted in place of the default values, thus saving time when building

new models. As an example, the process of setting up a simulation of a very simple system is outlined. Suppose that the output of a pulse generator is to be observed on an oscilloscope. First, the command *Simulink* is typed at the Matlab command window prompt in order to start Simulink and access the built-in Simulink libraries. A double-click on these basic libraries opens another window containing the related blocks. Figure 1 shows the basic Simulink libraries (which show upon typing the command *Simulink*). Figure 2 shows some of the blocks available in the Sources library.

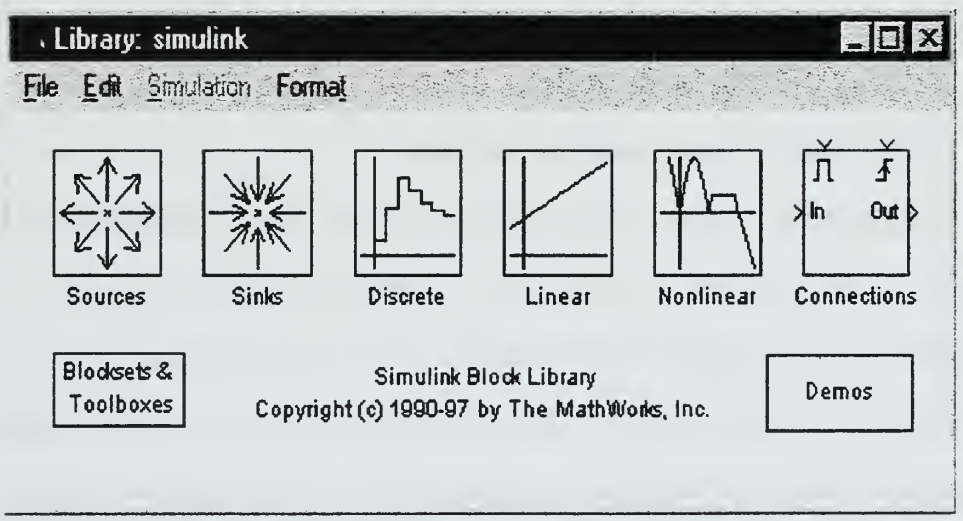

Figure 1. The Standard Simulink Block Library.

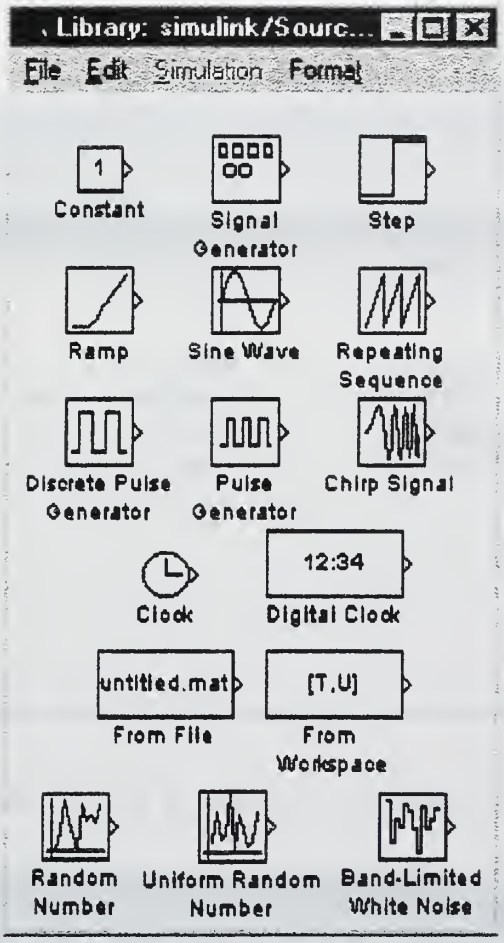

Figure 2. Blocks Available in the Sources Block Library.

To build the simple simulation the Sources sub-library needs to be opened. This displays yet another Simulink window from which the discrete pulse generator can be copied to a "New File" window. Similarly, by opening the Sinks sub-library, the oscilloscope block can be copied and placed into the new system window. Figure 3 shows the discrete pulse generator placed into a new window. Figure 4 shows the

oscilloscope block placed into the new window and finally Figure 5 shows the finished model, with the source connected to the oscilloscope.

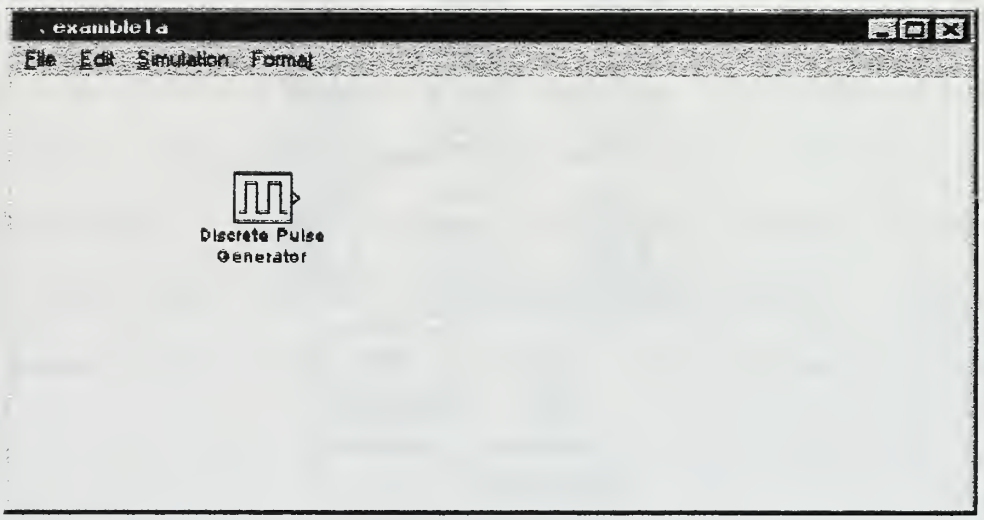

Figure 3. Placing the Discrete Pulse Generator Block.

Figure 4. Placing the Oscilloscope Block.

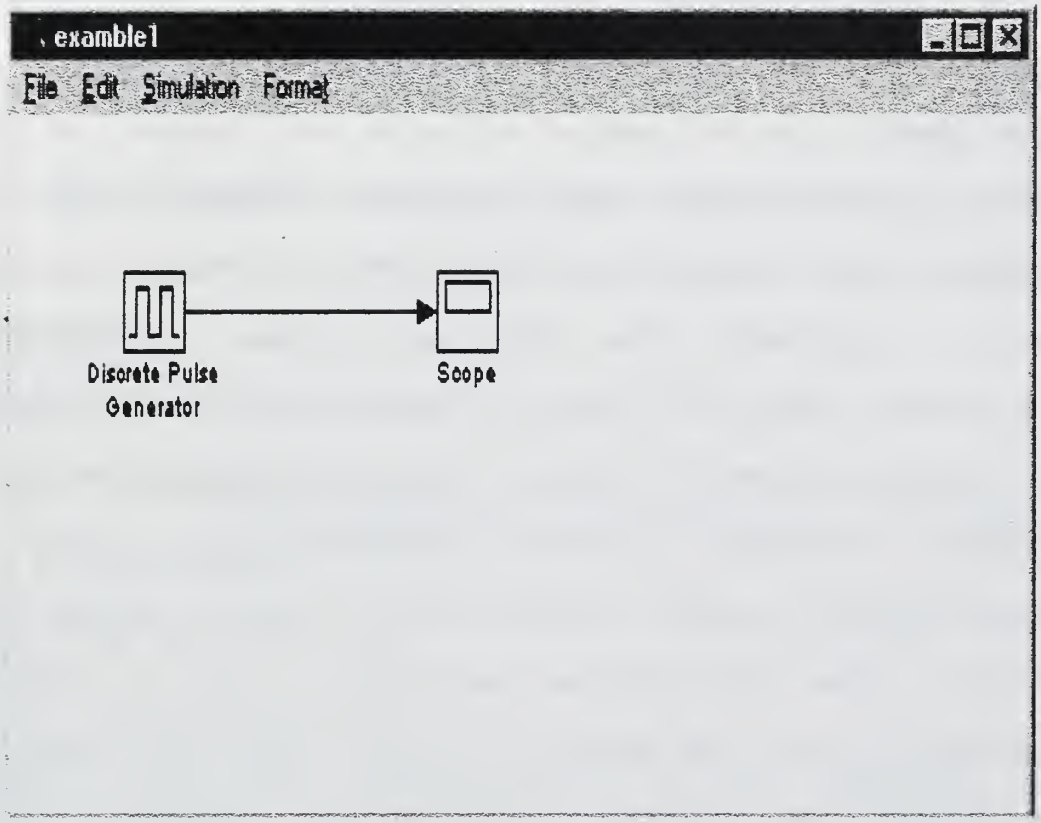

Figure 5. Completed Simulink Model.

All copied blocks maintain the original default internal parameters until modified by the user. Therefore, by simply connecting the sine wave "source" block to the oscilloscope "sink" block and selecting the parameters of the source, the system is ready to perform a simulation [Ref 1,2].

D. COMMUNICATIONS TOOLBOX

The Communications Toolbox is designed for use with Matlab and Simulink and represents a collection of computational functions and simulation blocks for research, development, system design analysis, and simulation in the communications area. The toolbox contains ready-to-use functions and blocks, which can be easily modified by the user to implement the required schemes, methods, and algorithms. Simulink blocks and Matlab functions accelerate the design process by helping users to rapidly develop and analyze different system designs. By using the Communications Toolbox (whose functions can be either called directly from the Matlab workspace and .m files or implemented as specialized Simulink blocks) time-domain simulation of various communication systems can be implemented. The toolbox adds a variety of Matlab functions and Simulink blocks that are very useful for communication system simulation. These include:

- Signal Generators (data source)
- Source Coding/Decoding
- Error-control Coding
- Baseband and Passband Modulation/Demodulation
- Transmitting and Receiving Filters

- Baseband and Passband Channel Models
- Multiple Access
- Synchronization
- Galois fields calculations. [Ref 1]

E. CONSTRUCTING A COMMUNICATIONS SYSTEM MODEL

A simple communication system contains these basic components:

- information source,
- the transmitter,
- the channel,
- the receiver, and
- information "sink."

Suppose that the performance of a coherent MFSK communication in the presence of additive white Gaussian noise needs to be determined. The Communications Toolbox is started by typing the command *commlib* at the Matlab command line prompt. This gives access to the "top level" communications library shown in Figure 6. The transmiting part of a communication system is comprised by the upper row of blocks while the receiving part is comprised of the bottom row of blocks.

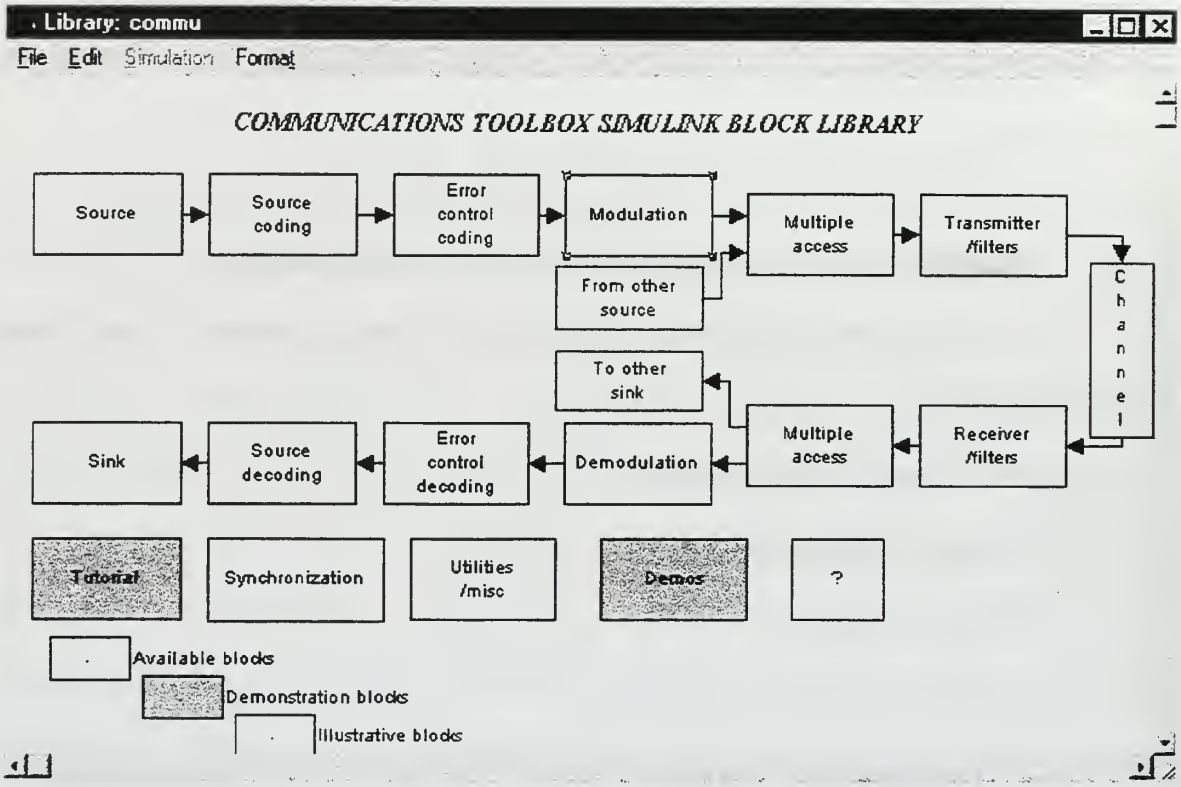

Figure 6. The Communications Toolbox Simulink Block Library.

Since in most cases the receiving computation is exactly the reverse of the transmitting computation, a double click on any of the blocks in this library opens a sub-library that includes both the transmitting and the corresponding receiving blocks. A sub-library typically contains function blocks used to build the models and demo blocks (cyan colored) that are used to demonstrate a block operation. Finally, by opening the source/sink library various types of sources become available. In order to build

an MFSK system the "Sampled Read from Workspace" block is copied from the Source sub-libarary and placed into a new Simulink window. The "Sampled Read from Workspace" block reads a row of data from the Matlab workspace at every sampling point. Similarly, by opening the modulation/demodulation library (Figure 7) and double clicking on the Digital Modulation/ Demodulation sub-library, the MFSK Mod Baseband and the Coherent MFSK Demod blocks can also be copied and placed into the new window.

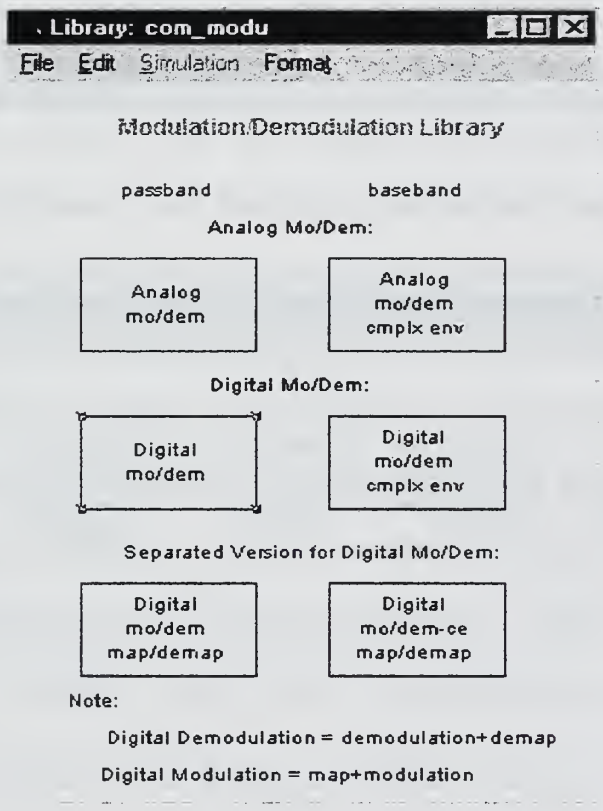

Figure 7. The Modulation/Demodulation Library.

The Coherent MFSK Demod block demodulates the input, which is a complex analog signal (because of the baseband implementation of the passband modulation) and is defined by five parameters:[M-ary Number, Tone Space, Symbol Interval, Initial Phase, Sample Time]. The MFSK Baseband Mod block accepts a scalar input in the range [0 M-1] and is defined by four parameters: [Tone space, symbol interval, initial phase, sample time].

From the channel sub-library the AWGN channel block is selected and copied into the new window. This block adds AWGN to the signal and is defined by three parameters. The first two specify the mean and the variance of the noise output and the third initializes the seed for the Gaussian random number generator. Finally, the error-rate meter block is copied from the sink sub-library. This block, shown in Figure 8, performs the symbol-to-symbol comparison between the sender and the receiver, counts symbol errors, and computes the symbol error rate. As shown in Figure 8, 50 symbols have been transmitted and the receiver has detected 13 symbols incorrectly. So the symbol error-rate is $\frac{13}{50} = 0.270833$. All the copied blocks are connected in order to complete the MFSK system shown in Figure 9.

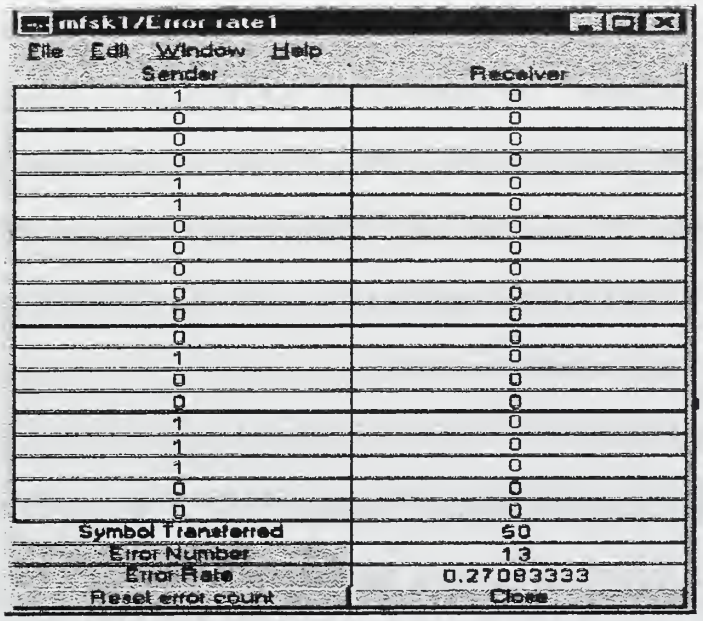

Figure 8. The Error-Rate-Meter block.

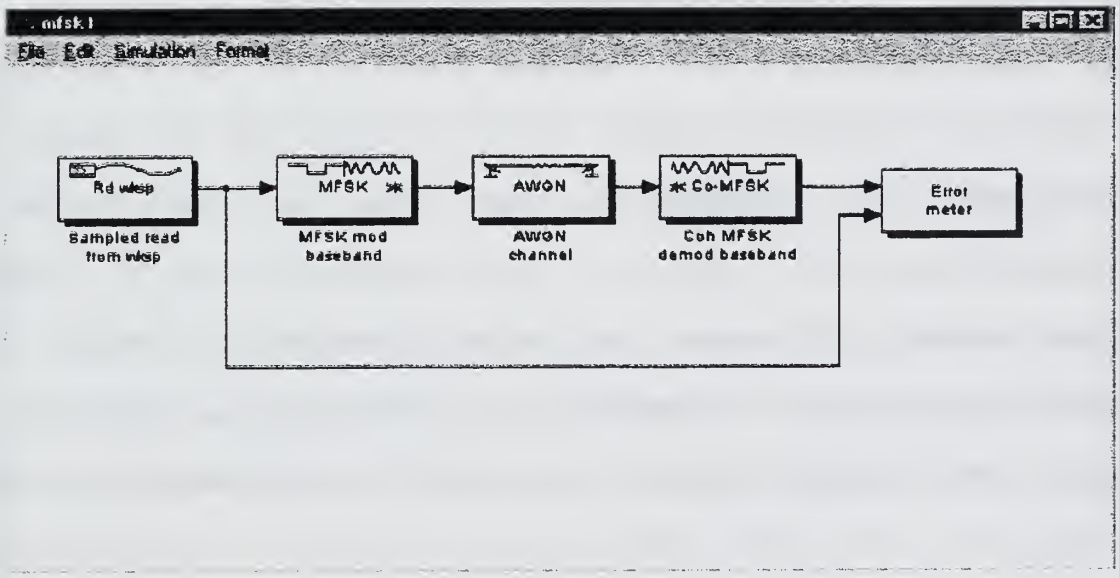

Figure 9. A Simple Model of a Coherent MFSK System.

F. MODEL ANALYSIS

Comunication system simulation can take place upon the completion of the communication system model. There are basically three different ways of implementing and controlling a simulation. The first is to run the simulation interactively from the menu bar and observe the signals at various inputs and outputs using oscilloscopes while the simulation is running. This method is easy to learn and simple to use, but it also incurs the highest overhead and is therefore the slowest to execute. The second option is to run the simulation by executing a .m file. The main advantage of this method is that it provides greater flexibility than the first method. For example, parameters in the blocks can be changed for repeated simulations and initial conditions for various blocks can be overridden. Running a simulation from a .m file allows one to change one parameter (or parameters) each time the simulation is repeated and the results of the simulation can be brought into Matlab workspace for further analysis. Finally, the third possibility of implementing a simulation (which is the most complex and flexibile) is based on the fact that every Simulink block has a Matlab equivalent as an S-function. Each such S-function has the same name as the corresponding model and can be called in a variety of ways to provide

information about the model it represents. The S-functions can provide information about the number of inputs, outputs, and states (both continuous and discrete) of the model as well as the derivatives of the outputs. All of the analysis tools provided with Simulink interact with models through S-functions. The linmod and trim .m files are examples, which use these functions. These are shown in Table 1.

	Modeling and trimming	Descreption
1	linmod	Generate linearized model of a continuous system
2	dlinmod	Generate linearized model of a with discrete elements.
3	trim	Trim systems about equilibrium point

Table 1. Partial List of Analysis Functions (from Ref [9]).

Generally speaking, the three ways to implement/run a simulation are not strictly delineated and different usage may be appropriate at different stages of model development [Ref 2,6,9].

Finally, Simulink provides a number of methods for the numerical solution of ordinary differential equations that

can be selected prior to simulation and changed even while the simulation is running. These numerical methods for solving "initial value problems" are shown in Table 2.

	Integration	Description
1	linsim	Linear systems method
2	rk23	2nd order Runge-Kutta method
3	rk45	4th order Runge-Kutta method
4	euler	Euler's method
5	adams	Adams predictor-corrector method
6	gear	Gear's predictor-corrector method for stiff systems

Table 2. Functions used for Simulation of Ordinary Differential Equations (from Ref. [9]).

III. MODEL OF A MFSK PASSBAND DIGITAL COMMUNICATION SYSTEM AND ITS BASEBAND EQUIVALENT

A. BASEBAND EQUIVALENCE FOR PASSBAND SIGNALS AND SYSTEMS

The most convenient way of representing a narrowband system for simulation purposes is to work with equivalent baseband quantities. Consider a communication system operating with 1 GHz and a bandwidth of 20 MHz. According to the Nyquist Sampling Theorem, we need a sampling rate of at least $2xf_{max}$ = 2x1,010 MHz to simulate this system. A sampling rate of this magnitude is not practical. Since the bandwidth of the baseband equivalent is several orders of magnitude smaller than the passband signal's center frequency (20 MHz versus 1 GHz), we can simulate the passband system by establishing its baseband equivalent system and simulating the equivalent system using a 20 Megasamples/second sampling rate instead of the over 2 Gigasamples/second sampling rate required for the passband signal. Therefore, simulating the baseband equivalent of a passband signal substantially reduces the required computer storage (memory and disk space) and the calculation time.

A (real) passband signal can be expressed in polar (amplitude and phase) form as:

$$x(t) = a(t)\cos[2\pi f_c t + \varphi(t)] \Rightarrow$$

$$x(t) = \mathrm{Re}\{a(t)\, e^{j\varphi(t)}\, e^{j2\pi f_c t}\} \qquad (3.1a)$$

where $a(t)$ is the envelope, f_c is the carrier frequency, Re is the real operator and $\varphi(t)$ is the phase of $x(t)$.

A (real) passband signal can be also expressed in Cartesian form:

$$x(t) = x_I(t)\cos(2\pi f_c t) - x_Q(t)\sin(2\pi f_c t) \qquad (3.1b)$$

where $x_I(t)$ is the in-phase component and $x_Q(t)$ is the quandrature component of $x(t)$.

The corresponding complex baseband (or lowpass) signal is defined as:

$$x_{LP}(t) = a(t)\, e^{j\varphi(t)} \qquad (3.2a)$$

or:

$$x_{LP}(t) = x_I(t) + jx_Q(t) \qquad (3.2b)$$

In order to transform the real passband signal of equation (3.1a) to the complex baseband signal of equation (3.2b), we suppress the carrier by dropping $e^{j2\pi f_c t}$.

The term $a(t)e^{j\varphi(t)}$ is called the complex envelope of $x(t)$. The baseband nature of $x_{LP}(t)$ and the low sampling rate required to satisfy Nyquist's Theorem for narrowband

signals is evident. Complex envelope analysis is very useful in analyzing the responses of passband systems to passband signals. Finding the response of passband systems to passband signals is one of the most essential tasks in communications system analysis.

If we denote the Fourier Transform of the passband signal $x(t)$ as $X(\omega)$ $\omega=2\pi f$ and the Fourier Transform of its complex envelope $x_c(t)$ as $X_c(\omega)$ then:

$$X(\omega)=0.5\ X_c(\omega-\omega_c) + 0.5\ X_c^*(-\omega-\omega_c) \qquad (3.3a)$$

where * represents the complex congugate. The first term represents complex envelope's baseband spectrum scaled by 0.5 and shifted to the carrier frequency. The second term represents the complex envelope's baseband spectrum scaled by 0.5, shifted to carrier frequency, folded about $\omega=0$ axis (to negative frequencies) and conjugated. Also, we can conclude that the first term represents the spectrum of $X(\omega)$ for positive frequencies while the second term represents the spectrum of $X(\omega)$ for negative frequencies.

Baseband equivalents of passband systems can be determined in a similar manner. For example, consider a filter with a passband frequency response $H(\omega)$ and real impulse response $h(t)$. We can now express the frequency

response $H(\omega)$ in terms of the baseband equivalent frequency response:

$$H(\omega)=0.5\ H_c(\omega-\omega_c)\ +\ 0.5\ H_c^*(-\omega-\omega_c) \qquad (3.3b)$$

From equations (3.3a) and (3.3b), we can conclude that the complex envelope of the passband response to a passband input can be obtained as the response of an equivalent baseband system to the complex envelope of the passband input $x(t)$. In this manner, the baseband equivalent system can be used to avoid calculations involving high frequency carriers [Ref 5,8].

B. MONTE CARLO SIMULATION

Symbol error probabilities (SEP) and bit error probabilities (BER) can be estimated using simulations with very large numbers of symbols transmitted by counting the symbol errors and taking the ratio of the number of symbol errors to the number of symbols as the number of symbols tends to infinity, assuming that the symbols are generated independently of one another. This is the essence of the Monte Carlo Simulation technique. The technique is simple but requires significant computing power and/or long simulation times when accurate estimates are needed for

small symbol/bit error rates. For example, if we count 10 errors, we can find from Figure 10 the SEP intervals that contain the true SEP with 90%, 95%, and 99% confidence [Ref 5].

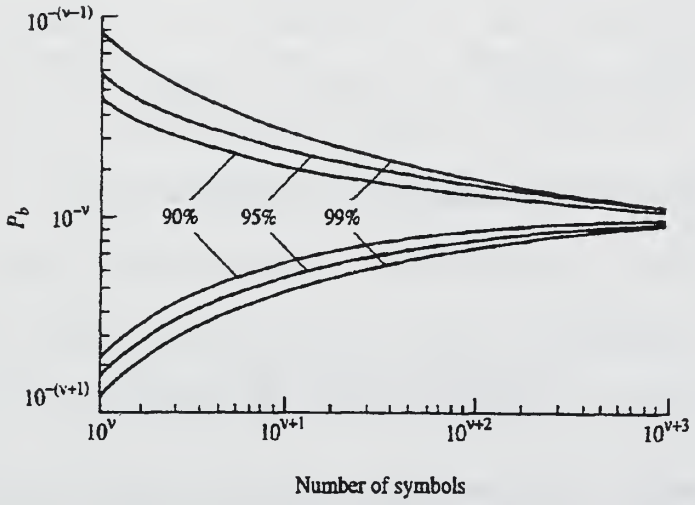

Figure 10. Confidence bands on P_b when observed value is $10^{-\nu}$ for the Monte Carlo technique (source: Jeruchim etal.,1984).

From Figure 10, if a symbol (or bit) error probability is 10^{-1} (ν=1), then for 100 experimental errors the actual ("true") probability of error is between 0.1 to 0.065 and 0.1 to 0.3 with 95% confidence. However, if errors are not independent from each other (errors occur in a "burst"), the

first error in a burst is more informative of the error statistics than all subsequent errors in the same burst. Therefore, in the "dependent error" case we need to count a larger number of errors for the same SEP confidence levels, which means even longer simulation times [Ref. 5]. If the number of experimental errors is denoted η_e and the number of the received bits (or symbols) is denoted N_b, then the confidence with which the BER is upper and lower bounded by $\frac{\eta_U}{N_b}$ and $\frac{\eta_L}{N_b}$, respectively, is given by

$$\text{Prob}\{\frac{\eta_L}{N_b} \le \text{BER} \le \frac{\eta_U}{N_b}/\eta_e\} = \Theta \qquad (3.4)$$

where $\Theta(0<\Theta<1)$ is the confidence level and η_L, η_U is the confidence interval. For a given number of experimental errors and if certain reasonable conditions are fulfilled (effects of disturbances are independent for each error and error-count statistics can be approximated as Gaussian distributed), then the lower and upper number bounds for the errors with 95.4% confidence interval are given from the following equations:

$$\eta_L=\eta_e + 2-2\sqrt{\eta_e+1} \qquad (3.5)$$

$$\eta_U = \eta_e + 2 + 2\sqrt{\eta_e + 1} \qquad (3.6)$$

where η_e is the number of experimental errors and η_L, η_U is the confidence interval. Generally speaking, error bounds converge slowly to each other (that is the estimates become more accurate) as the number of observed experimental errors η_e increases. For ten experimental errors, the true number of errors is between 5.4 and 18.6 (between approximately -50% and +76% of the observed value), with 95.4% confidence, while for 100 experimental errors the true results will be between 81.9 and 122 (between approximately -18% and +22% of the observed value) also with 95.4% confidence. The values of η_L and η_U from (3.5) and (3.6), respectively, and their ratio are tabulated in Table 3 for several values of η_e and the confidence level $\Theta = 0.954$ (95.4% confidence)[Ref.6].

	Experimental errors	Lower limit of true errors (η_L)	Upper limit of true errors (η_U)	Ratio $\frac{\eta_L}{\eta_U}$
1.	5	2.1	11.9	5.7
2.	10	5.4	18.6	3.5
3.	20	12.8	31.2	2.4
4.	50	37.7	66.3	1.8
5.	100	81.9	122	1.5
6.	200	174	230	1.3

Table 3. 95.4% Confidence Intervals for Various Number of Experimental Errors (from Ref. [6]).

C. M-ARY FREQUENCY SHIFT-KEYING(MFSK)

M-ary frequency-shift keying (MFSK) modulation does not require phase coherent detection. This means that MFSK can be used in applications where the phase shift incurred by the transmission channel varies rapidly, as is the case, for example, in communications with high-speed aircraft. With MFSK modulation, each symbol is assigned a specific frequency and an MFSK signal can be described by the following equation:

$$S_{MFSK}(t) = \sum A\, p(t-nT)\, \cos[2\pi(f_0+a_n\Delta f) + \varphi_n] \qquad (3.7)$$

where:

Δf is the spacing between any two adjacent frequencies:

$$\Delta f = f_n - f_{n-1} \tag{3.7a}$$

a_n is determined by the information sequence and takes M integer values

f_0 is the lowest frequency

A is the amplitude of the MFSK signal

φ_n is the phase angle

p(t-nT) is the unit-amplitude rectangular pulse.

1. Bit Error Probability For Coherent MFSK

In order to obtain the exact expression for the MFSK bit error probability, the exact expression for the MFSK symbol error probability must be first derived. The coherent MFSK symbol error probability is given by the following expression [Ref 7]:

$$P_{e(sym)} = 1 - \frac{1}{\sqrt{2\pi}} \int_{\sqrt{2SNR_{sym}}}^{\infty} (1 - Q(\zeta))^{M-1} e^{-(\zeta - \sqrt{2SNR_{sym}})^2} d\zeta \tag{3.8}$$

where:

$$\zeta = z/\sigma \tag{3.8a}$$

σ^2 is the variance of wideband Gaussian noise

M is the number of symbols

SNR_{sym} is the symbol SNR

$$SNR_{sym} = SNR_{bit}\frac{\log(M)}{\log(2)} \tag{3.8b}$$

and $Q(\zeta)$ is the Q-function given by the following:

$$Q(x) = \frac{1}{2}(1-\text{erf}\left(\frac{x}{\sqrt{2}}\right)) \tag{3.8c}$$

where the error function (erf) is defined as

$$\text{erf}(z) = \frac{1}{\sqrt{2\pi}}\int_0^z e^{-x^2}\,dx \tag{3.8d}$$

The following auxiliary functions are introduced for convenience:

$$\gamma(M, SNR_{bit}) = \sqrt{2\frac{\log(M)}{\log(2)}SNR_{bit}} \tag{3.9}$$

$$\psi(\zeta, M, SNR_{bit}) = \frac{1}{\sqrt{2\pi}}(1-Q(\zeta))^{M-1}e^{\frac{-(\zeta-\gamma(M,SNR_{bit}))^2}{2}} \tag{3.10}$$

By using these auxiliary functions we now can express the bit error probability [Ref 7] as a function of the number of symbols M and the bit signal-to-noise ratio SNR_{bit}:

$$P_{e(bit)} = \frac{1}{2}\frac{M}{M-1}\left[1-\int_{\gamma(M,SNR_{bit})-5}^{\gamma(M,SNR_{bit})+5} \psi(\zeta, M, SNR_{bit})d\zeta \right. \qquad (3.11)$$

2. Bit Error Probability For Noncoherent MFSK

In order to obtain the exact expression for the MFSK bit error probability, the exact expression for the MFSK symbol error probability must be first derived. The symbol error probability for noncoherent MFSK is given by the following expression [Ref7]:

$$P_{e(sym)} = 1-\sum_{k=0}^{M-1}\frac{(-1)^K(M-1)!}{(k+1)k!(M-1-k)!}e^{\frac{-k}{k+1}\cdot\left(\frac{A^2.T}{2.N_0}\right)} \qquad (3.12)$$

where:

M is the number of symbols, $M = 2^k$, k is an integer,

$\frac{N_0}{2}$ is a two-sided power spectral density, $N_0 = \sigma^2 4T$, and

A is the amplitude of the MFSK signal.

We can also express the noncoherent MFSK symbol error probability in terms of the symbol energy:

$$P_{e(sym)} = 1-\sum_{k=0}^{M-1}\frac{(-1)^K(M-1)!}{(k+1)k!(M-1-k)!}e^{\frac{-k}{k+1}(\frac{E_{sym}}{N_0})} \quad (3.13)$$

or in terms of the symbol signal-to-noise ratio:

$$P_{e(sym)} = 1-\sum_{k=0}^{M-1}\frac{(-1)^K(M-1)!}{(k+1)k!(M-1-k)!}e^{\frac{-k}{k+1}SNR_{sym}} \quad (3.14)$$

The symbol error probability for noncoherent MFSK is related to the bit-error probability for $M = 2^k$ by [Ref7]:

$$P_{e(bit)}(M, SNR_{bit}) = \frac{1}{2}\frac{M}{M-1}[1-\sum_{k=0}^{M-1}\frac{(-1)^K(M-1)!}{(k+1)k!(M-1-k)!}e^{\frac{-k}{k+1}(\frac{\log(M)}{\log(2)})SNR_{bit}}] \quad (3.15)$$

or

$$P_{e(bit)}(M, SNR_{bit}) = \frac{1}{2}\frac{M}{M-1}[\sum_{k=1}^{M-1}\frac{(-1)^K(M-1)!}{(k+1)k!(M-1-k)!}e^{\frac{-k}{k+1}(\frac{\log(M)}{\log(2)})SNR_{bit}}] \quad (3.16)$$

D. MFSK MODELS AND BLOCK ANALYSIS

1. Coherent MFSK Model (Noise Only)

The following figure shows the coherent MFSK (noise case only) model, which has been used for the simulation.

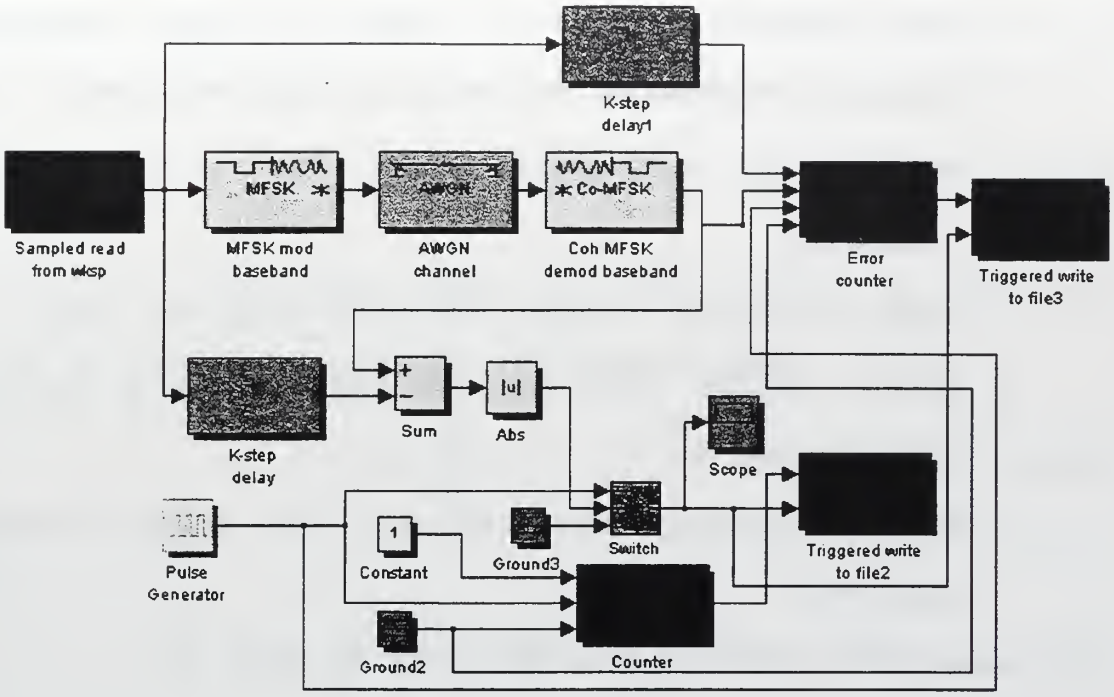

Figure 11. Coherent MFSK Model for Noise Only Case.

2. Block Analysis

a) Coherent MFSK Mod Baseband

Four parameters define the coherent MFSK Mod Baseband block:

[Tone space, symbol interval, initial phase, sample time].

The MFSK Baseband Mod block accepts a scalar input in the range [0, M-1]. (For example, M in the coherent BFSK model is 2). The output of the MFSK Mod block is a unit-amplitude FSK modulated analog signal. The following must be specified for the MFSK Mod block:

- Tone space (Δ_f) (also known as the frequency separation between two neighboring frequencies and defined as Δ_f=1/(2Ts), where T_s is the symbol interval).
- Symbol interval T_s (also the data sampling time)
- Initial phase (the initial phase of the carrier, commonly set to 0)
- Sample frequency (the inverse of the signal sampling interval)

The theoretical BER for coherent BFSK is given as:

$$P_B = Q\left(\sqrt{\frac{E_B}{N_O}}\right) \tag{3.17}$$

where E_B is the average bit energy and N_O is the single-sided noise power spectral density.

From the Nyquist Theorem, at least two samples per shortest period are needed in order to represent a signal. For the baseband equivalent model of binary FSK (BFSK), one symbol is represented as DC and the other symbol as a carrier with the frequency Δf. We have selected the symbol interval T_s as 1 second. (In the binary case this is also the bit interval T_b). For a good representation of the signal we sample at twice the Nyquist rate; that is, four

samples per bit interval, or four samples per second, are provided. The number of samples per second N_s is given as:

$$N_s = \frac{T_b}{\Delta_t} \tag{3.18}$$

where Δ_t is the MFSK signal (not data) sampling time. For BFSK and sampling at twice the Nyquist rate, we get:

$$\Delta_t = \frac{T_b}{N_s} \tag{3.19}$$

or

$$\Delta_t = 1\text{sec}/4\text{samples per sec} \Rightarrow \Delta_t = 0.25\text{Hz}.$$

For orthogonal MFSK and coherent detection, the frequency separation must be an integer multiple of $T_s/2$. The lowest separation for orthogonality with coherent detection is:

$$\Delta_f = \frac{1}{2}T_s \Rightarrow \Delta_f = \frac{1}{2}T_B \Rightarrow \Delta_f = \frac{1}{2} \Rightarrow \Delta_f = 0.5\text{Hz} \tag{3.20}$$

This value is defined as the "tone space" (in Hz) in the Simulink model. In the case of a coherent 4FSK or 8FSK signal the sample time Δt must be changed because the highest frequency in the spectrum is approximately $(M-1)\Delta f$ since the maximum frequency for MFSK is defined as $f_{max} = (M-1)\Delta_f$. In order to have a good representation of the signal,

we transmit four samples per symbol interval (or four samples per second since we have selected T_s=1 second), and the sampling frequency for MFSK is:

$$f_{sampling} = 4(M-1)\Delta f \qquad (3.21)$$

The sampling time Δt is the reciprocal of the sampling frequency:

$$\Delta t_{sampling} = 1/f_{sampling} \qquad (3.22)$$

b) Coherent MFSK Dem Baseband

This block demodulates the input, which is a modulated complex analog signal. Its output is an integer in the range [0,M-1]. Five parameters define the Coherent MFSK Dem Baseband block:

- [M-ary number, tone space, symbol interval, initial phase, sample time]

The above parameters should match the ones used in the corresponding Coherent MFSK Mod Baseband block. We have selected the following values:

- Tone space (Δ_f): $\Delta_f = \frac{1}{2} T_s$, where T_s is the symbol interval
- Symbol interval (T_s): 1 sec.
- Initial phase: 0

- Sample time : $1/(4(M-1)\Delta f)$
- M-ary number: 2, 4, or 8

c) AWGN Channel

This block adds AWGN to the signal being transmitted through this channel. Three parameters define the AWGN channel. The first two specify the mean and the variance of the noise output. The third one initializes the seed. The seed effectively selects a pseudorandom sequence to be generated (for the same value of the seed the same sequence is always generated).

d) Symbol Counter

The counter block counts the number of symbols. There are three input ports to the counter. The first port inputs a constant value of 1. The second port inputs the clock signal, which has one pulse per each symbol generated, and the third input port is unused ("grounded"). The value of 0.5 is set as threshold. Each time the clock pulse crosses the threshold going from 0 to 1 (on the rising edge of the clock pulse) the counter increments its output by 1, thus effectively counting the input symbols.

e) Error Counter

This block compares the delayed "original" signal (first input) and the demodulated signal (second input) at the end of each symbol interval. The timing is controlled by the third input when the rising edge of the square wave (from the pulse generator) crosses the threshold. The fourth input is not used ("grounded"). The output of this block increases by one each time the absolute value of the difference between the inputs 1 and 2 exceeds the specified tolerance, that is each time the detected symbol differs from the original (sent) symbol. Therefore, by setting the tolerance of the input difference to 0 (or a very small non-zero value), the symbol errors are counted.

f) Triggered Write To File

Four parameters define the Triggered Write to File block:

[File name, data type, number of trigger pulses between saved data, threshold in detecting trigger signal]

This block writes a record to a file only at the rising edge of the signal coming from the pulse generator (the block can also isolate and write all the errors to a file in case of burst errors). The file can be an ASCII text file or a data type ("integer," "float"). The third parameter (number of trigger pulses between saved data) determines the number of

rising edges that should be received between file writes. Setting this field to zero forces the block to write a record at every rising edge of the trigger. This block has two input ports. The first input is the error counter value that is recorded into a file and the second input input is the "write trigger" signal from the pulse generator.

g) Triggered Write To Workspace:

Six parameters define the Triggered Write to Workspace block:

[Workspace variable, data type, number of trigger pulses between saved data, maximum row number, section to keep if overflow occurs, threshold in detecting trigger signal]

The saved workspace variable can be either a column vector when the first input port (message signal) has a scalar input or a matrix when the first input has a vector input. The first element of the input signal vector is saved in the first column; the second element of the input signal vector is saved in the second column, and so on. The block can save the input as string variables or data. The maximum row of the output variable is limited by a pre-defined number, which can be changed during simulation. After the limit is reached, the block keeps the portion of the data defined by the "overflow" entry. Data may be written into the workspace at every trigger or a number of triggers may be skipped

between sucessive writes, depending on the value of the corresponding entry (zero forces a write at every trigger).

h) K-Step Delay

This block implements a delay that is equal to an integer multiple of the sampling interval that is in effect for the particular simulation.

i) Sum

This block adds or subtracts the two inputs depending on the sign selection for the inputs: ++, +-, -+, --. Since the objective is to compare the original and the detected symbols, either +- or -+ can be used.

j) Abs

This block's output is the absolute value of the input. This block creates a positive value of 1 each time the original and the detected symbols differ, regardless of the sign (polarity) of the difference.

k) Pulse Generator

Four parameters define the pulse generator block: [Pulse period (symbol period), duty cycle (% of period), amplitude, start time]

This block generates a unit-amplitude square wave with the period equal to the symbol interval. The pulse generator

starts at the end of the first symbol interval such that the rising edges of the square wave coincide with the end of symbol times. Therefore, since there is a rising edge of the square wave at the end of each symbol interval, there is always a rising edge coincident with a symbol decision time.

l) Switch

The switch block has three input ports. The first port inputs the clock signal and the second input port inputs the absolute value of the sum of the delayed and demodulated signal. The third input is unused ("grounded").

m) Sampled Read From Workspace

The Sampled Read From Workspace block reads a matrix from MATLAB workspace. In our case the matrix is a matrix of pseudo-random integers generated using MATLAB's random integer function *randint*. There are four argument fields, delineated by commas, in the specification of the *randint* function. The first two define the size (rows and columns) of the matrix to be generated. The third field accepts either a single integer (the number of symbols M) or two-integer elements array [0, M-1]. In either case the range of random integers (the elements of the random matrix) is from 0 to M-1 and all M values are equally probable. The

last field is the seed for the pseudorandom matrix (for the same value of the seed the same matrix is always repeated).

Randint [number of rows, number of columns, [0 number of symbols -1], seed]

or

Randint [number of rows, number of columns, number of symbols, seed]

In the case of noncoherent MFSK (with noise only) the same model as for the coherent MFSK (noise case only) is used with only one difference. The coherent MFSK Mod Baseband and coherent MFSK Dem Baseband blocks are replaced with the noncoherent MFSK Mod Baseband and noncoherent MFSK Dem Baseband blocks. These blocks are defined by the same parameters as in the coherent MFSK Mod Baseband and coherent MFSK Dem Baseband blocks. Also, the frequency separation is $\Delta f = 1/T_s$ instead of $1/(2T_s)$.

3. Coherent MFSK Model For MFSK With Interference And Additive Noise

The coherent MFSK model with interference AWGN is shown in Figure 12.

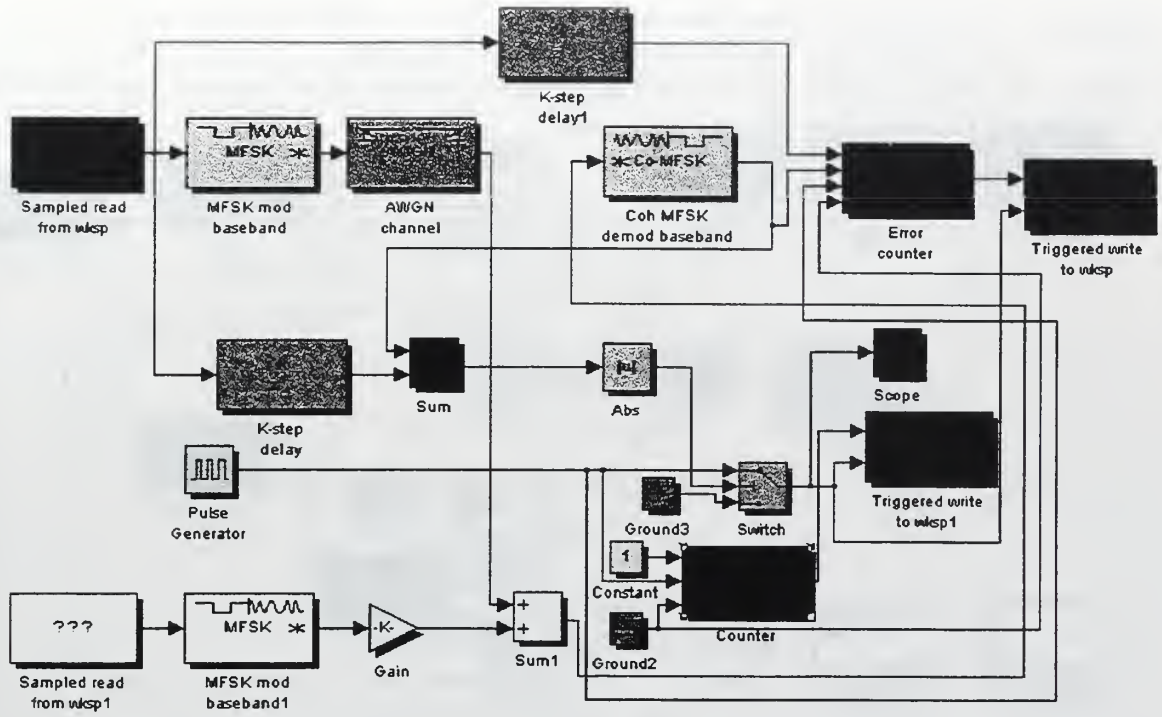

Figure 12. Model for Coherent MFSK with Interference and Additive Gaussian Noise.

The difference between this model and the model for coherent MFSK with noise only is that the white blocks representing the interference signal have been added. The interference signal is assumed synchronized with the desired signal. That is, symbol transitions of the interference occur at the same instants in time as the symbol transitions of the desired signal. For noncoherent MFSK with interference and AWGN, the same model is used (Figure 13) with the coherent MFSK Mod Baseband and coherent MFSK Dem

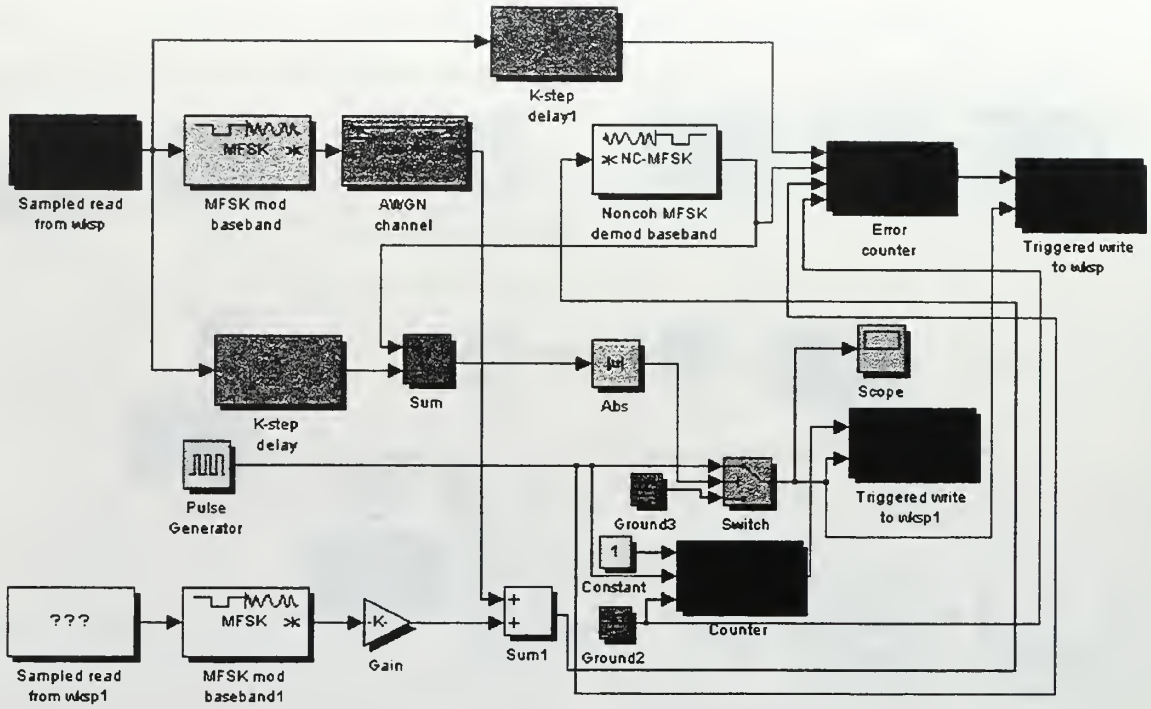

Figure 13. Model for Noncoherent MFSK with Interference and Additive Gaussian Noise.

Baseband blocks replaced with the noncoherent MFSK Mod Baseband and noncoherent MFSK Dem Baseband blocks and the frequency separation increased to $1/T_s$. In order to run multiple simulations for a number of different values of parameters such as the signal-to-noise ratio (SNR), Matlab .m files were used. For coherent MFSK, the Matlab programs sun_prep1.m and sun_fsk1.m (listed in the Appendix) were used. These programs can also be combined into a single program. For noncoherent MFSK, essentially the same Matlab

.m files were used, with the appropriate variables changed. The .m files are used to complete the three basic tasks: data generation and input, simulation, and recording the output into files, as shown in Figure 14.

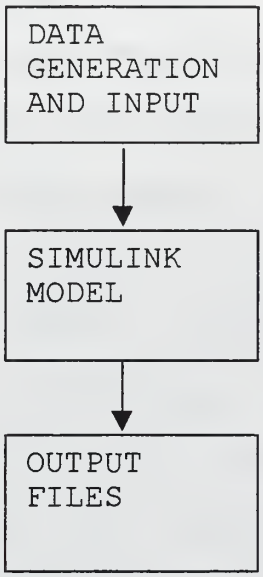

Figure 14. Flow Chart for Matlab .m File Simulation Implementation.

The first task involves the choices concerning the case we want to test (noise or noise and interference), the type of MFSK (2FSK,4FSK,8FSK…), and the simulation parameteres such as symbol duration T_s, oversampling factor, minimum and maximum signal-to-noise and signal-to-interference ratios, mimimum number of errors acceptable, the factor multiplying the number of errors (to select the number of symbols to

try), the maximum size of the random integer array, and the name of the file for saved data. The second task is the actual simulation started by the appropriate Matlab command and using the selected Simulink model (coherent MFSK or noncoherent MFSK). The third task is recording of the results: the signal-to-noise and signal-to-interference ratios, the number of errors, the number of symbols, and symbol error probability, all as ASCII data files.

IV. SIMULATION, ANALYSIS, AND PERFORMANCE VERIFICATION FOR AWGN

A. PERFORMANCE VERIFICATION FOR AWGN

In this chapter computer simulation results are presented in order to verify the performance of MFSK communication systems in the presence of AWGN. These results are the average of three computer simulation runs for each signal-to-noise (SNR) ratio. The value of two has been used as an oversampling factor (sampling rate of twice the minimum value). The simulations ran until at least 100 errors were observed. The bit error probability intervals for the 90%, 95%, and 99% confidence levels may be found from Figure 10 (Chapter III). For a given number of experimental errors and for the confidence level of 95.4%, the lower and upper bounds on the number of errors can be found from Table 3-1. The observed error number is a random variable, subjected to statistical scatter arising from different realizations (in general, a different number of errors is observed each time a simulation is run with a different set of seeds for data and noise). In order to prevent 'out of memory' errors, the data sequences were limited to 10^6 symbols for each simulation. If less than 100

errors were observed in a simulation, then the sequence was repeated until a sufficient number of errors occurred.

1. Results For Coherent Binary FSK (BFSK)

The theoretical and the experimental bit error probability for coherently detected binary FSK (BFSK) as a function of the SNR in dB is shown in Figure 15.

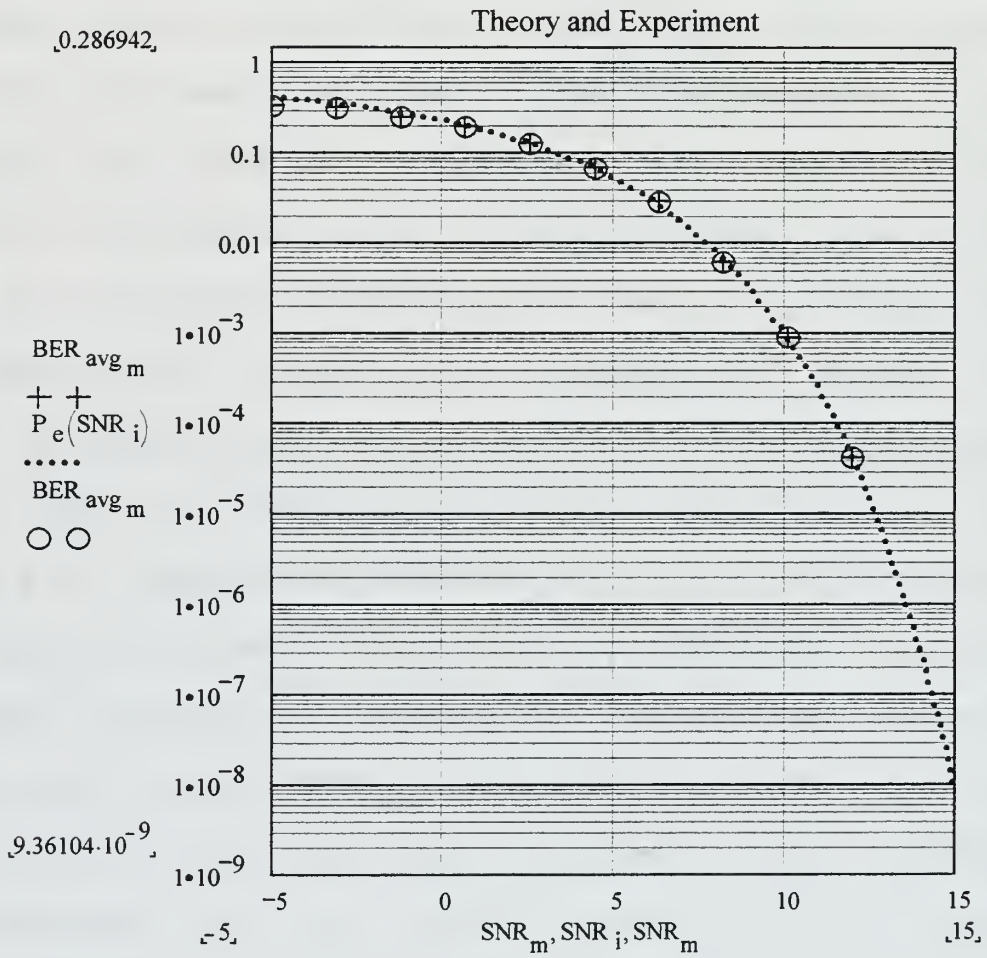

Figure 15. Probability of Bit Error (Theory and Simulation) for Coherent BFSK Detection.

The relative difference between the theoretical probability of error for coherent BFSK and the probability of error estimates obtained by the simulation are shown in Figure 16.

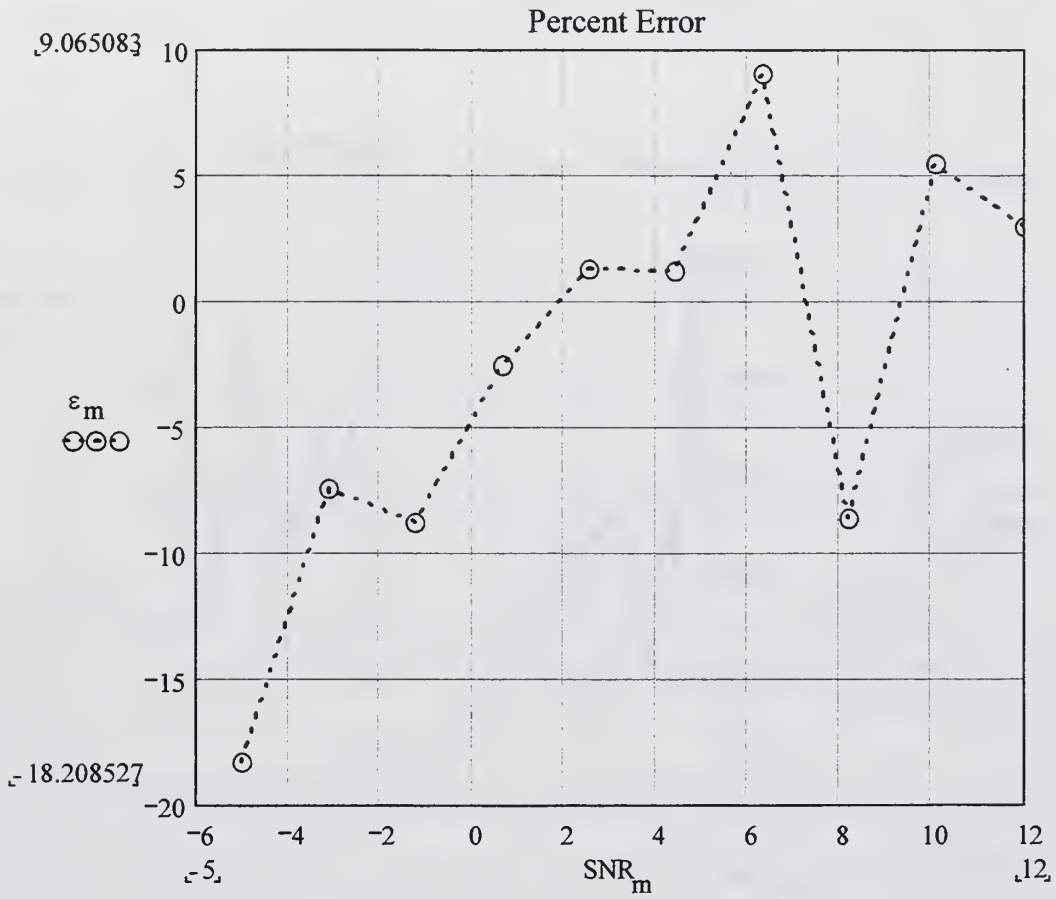

Figure 16. Percent difference between Theoretical and Simulation Results for Coherent Detection of BFSK.

The convergence of the estimate of the bit error probability as a function of the number of symbols transmitted in the course of the simulation is shown in

Figure 17. This allows us to estimate the minimum number of symbols required for a particular simulation and to verify that minimum has been reached. For this particular simulation, convergence is achieved at approximately $1.2 \cdot 10^5$ symbols.

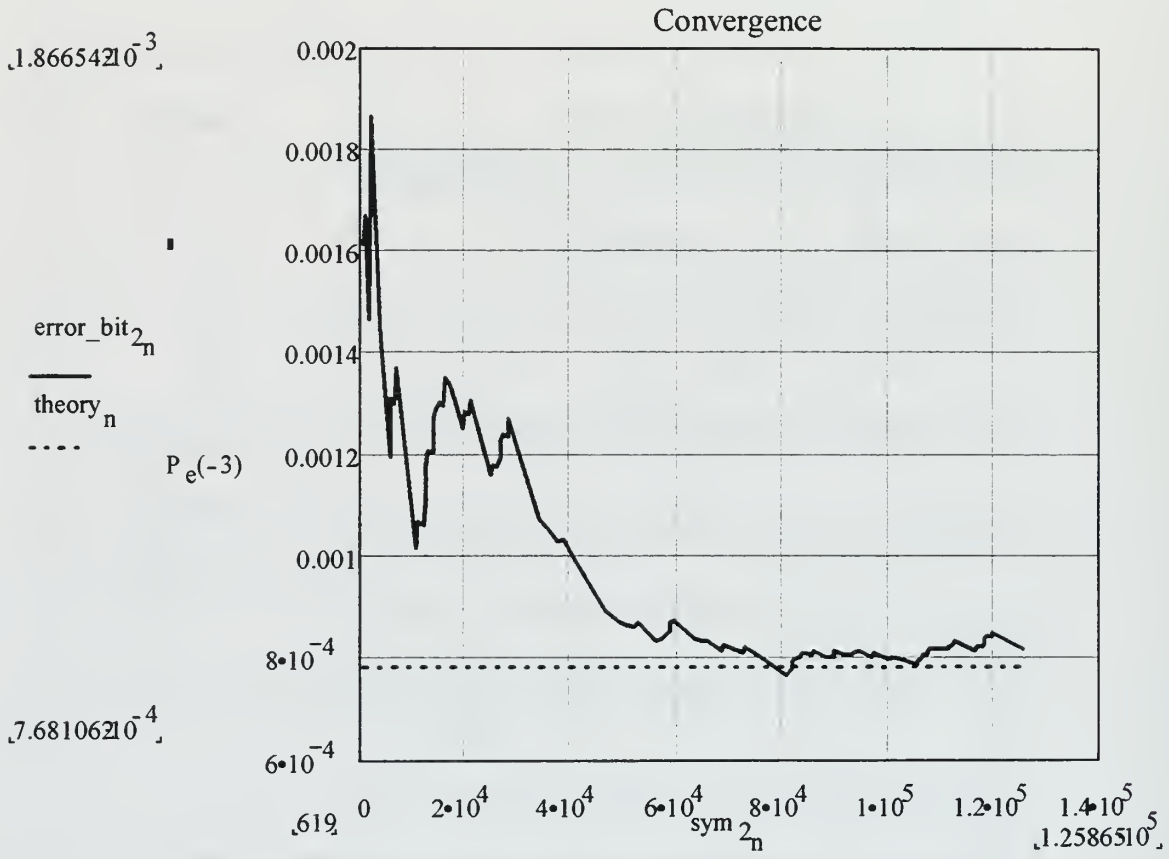

Figure 17. Convergence Plot for Coherent Detection of BFSK.

The accuracy of the simulation estimate for the bit error probability for coherent BFSK is shown in Figure 18. The convergence to zero as the number of symbols increases is evident.

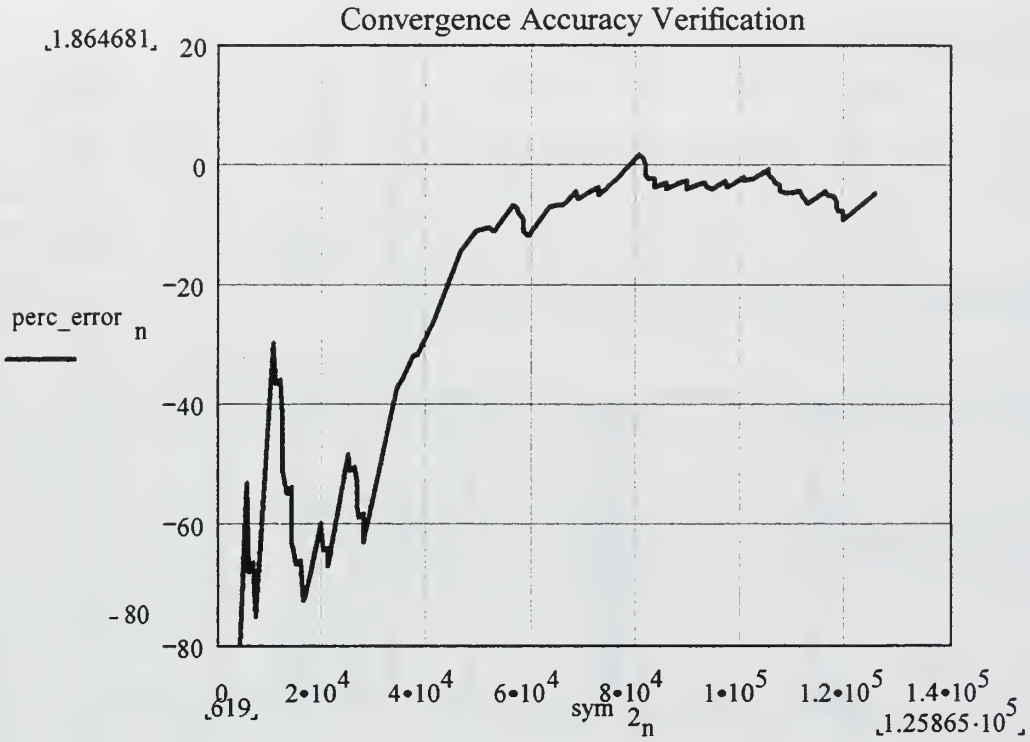

Figure 18. Percent Difference Between Theory and Simulation for Coherent BFSK as a function of the Number of Symbols.

The differences between the theoretical bit error probabilities for coherent 2FSK and the simulation estimates for various SNR's are plotted in Figure 19 as a histogram. All estimates are within the -20% to +10% interval about the

theoretical values, with most differences very close to 0 (approximately +2% off).

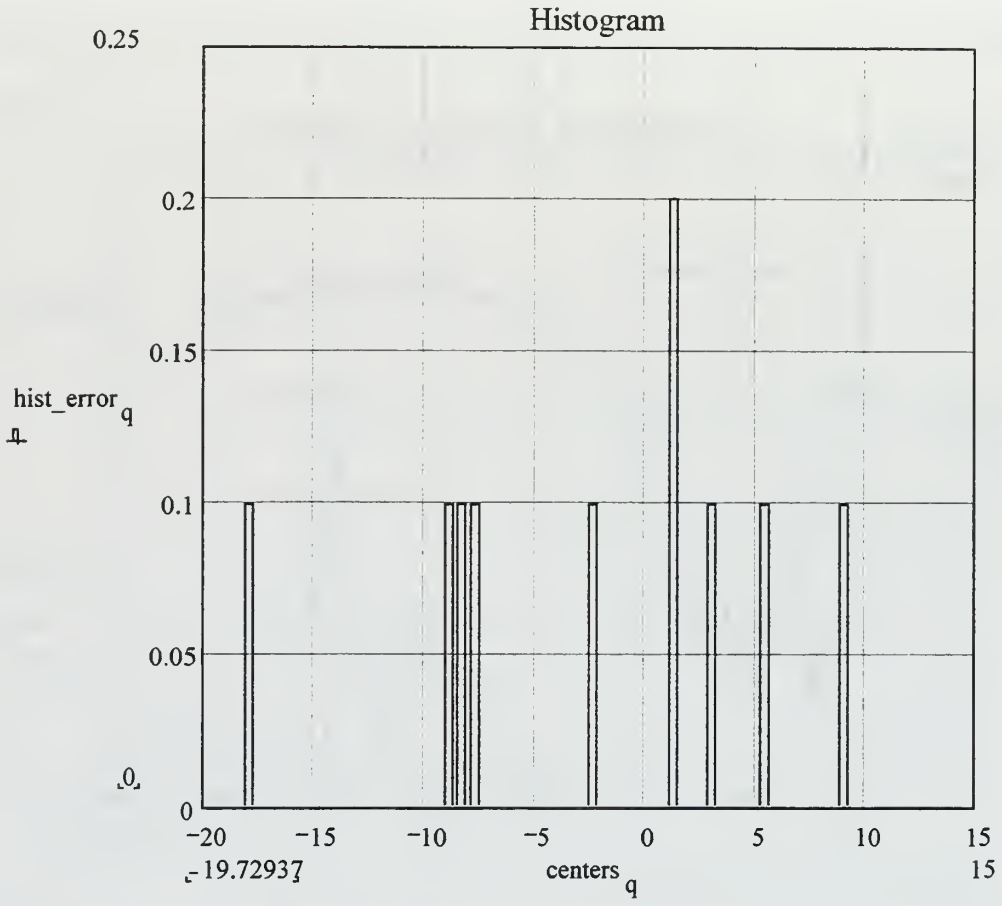

Figure 19. Histogram of Simulation Errors for Coherent BFSK.

2. Results For Coherent 4FSK

The theoretical and the experimental bit error probability for coherently detected 4FSK as a function of the SNR in dB is shown in Figure 20.

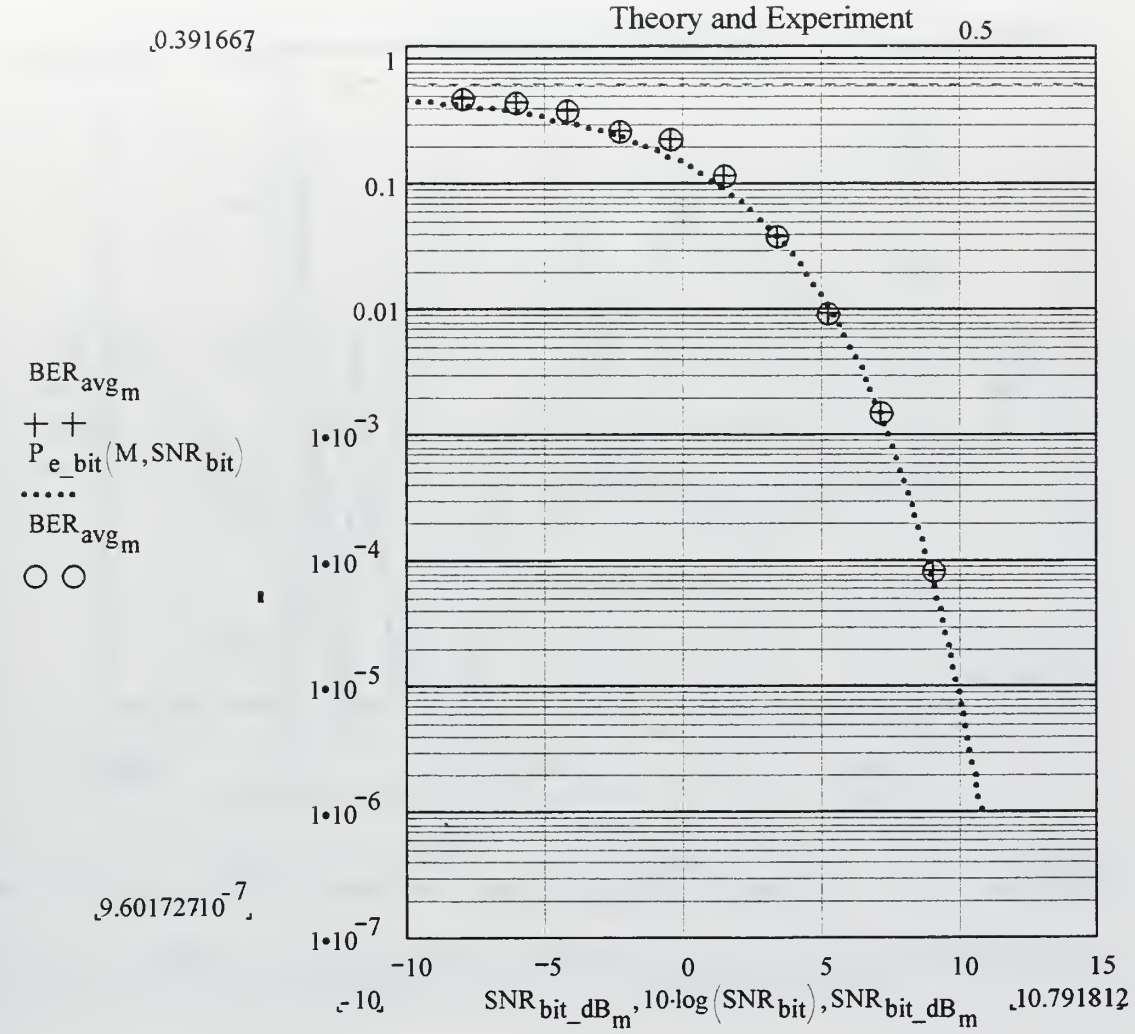

Figure 20. Probability of Bit Error (Theory and Simulation) for Coherent 4FSK Detection.

The relative difference between the theoretical probability of error for coherent 4FSK and the probability of error estimates obtained by the simulation is shown in Figure 21.

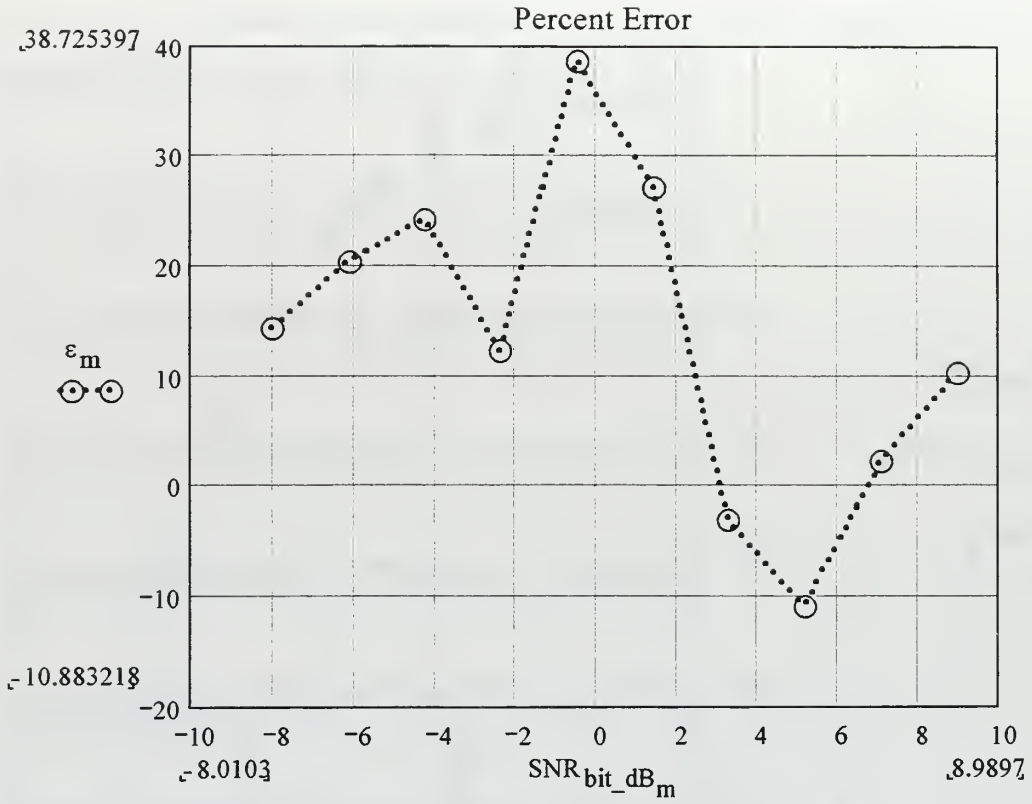

Figure 21. Percent difference between Theoretical and Simulation Results for Coherent Detection of 4FSK.

The convergence of the estimate of the bit error probability as a function of the number of symbols transmitted in the course of the simulation is shown in Figure 22. This allows us to estimate the minimum number of symbols required for a particular simulation and to verify that the minimum has been reached. For this particular simulation, the convergence is evident at approximately $4.5 \cdot 10^4$ symbols.

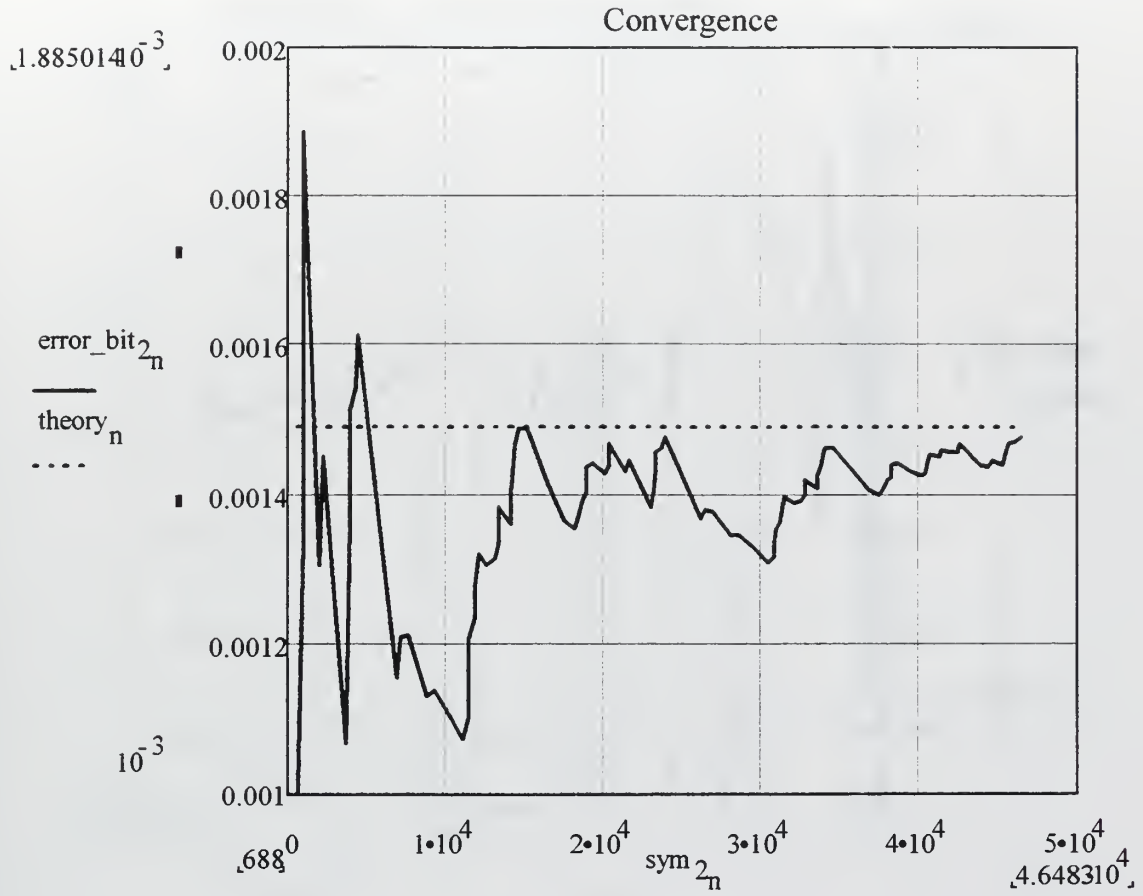

Figure 22. Convergence Plot for Coherent 4FSK.

The accuracy of the simulation estimate for the bit error probability for coherent 4FSK is shown in Figure 23. The 'oscillatory' convergence to zero as the number of symbols increases is evident.

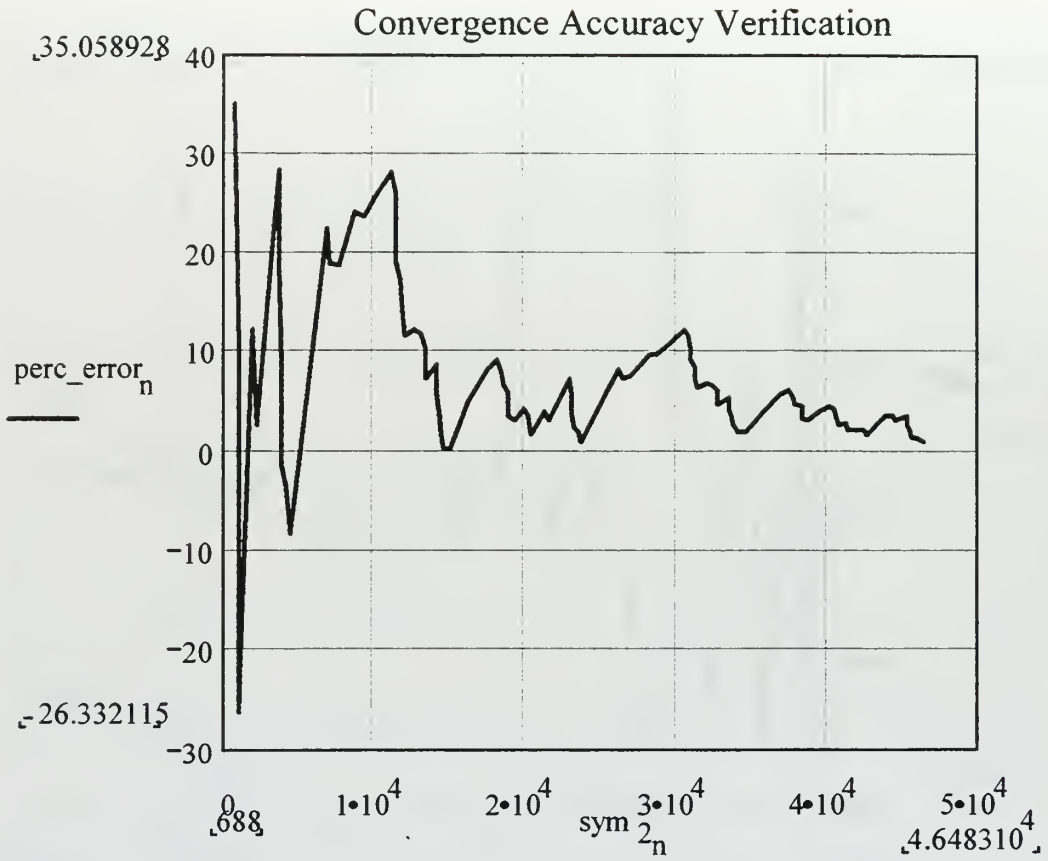

Figure 23. Percent Difference Between Theory and Simulation for Coherent 4FSK as a function of the Number of Symbols.

The differences between the theoretical bit error probabilities for coherent 4FSK and the simulation estimates for various SNRs are plotted in Figure 24 as a histogram. All estimates are within the -12% to +40% interval about the theoretical values, with most differences between +10% and +30%.

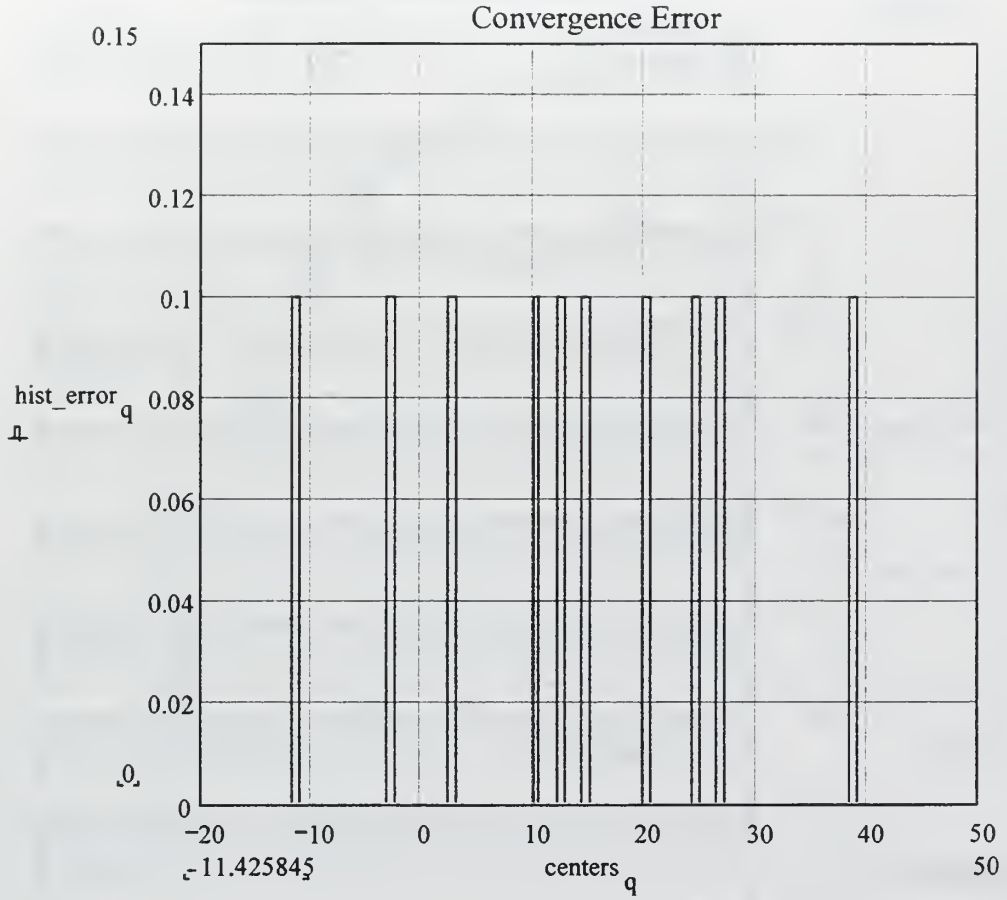

Figure 24. Histogram of Simulation Errors for Coherent 4FSK.

3. Results For Coherent 8FSK

The theoretical and the experimental bit error probability for coherently detected 8FSK (8FSK) as a function of the SNR in dB is shown in Figure 25.

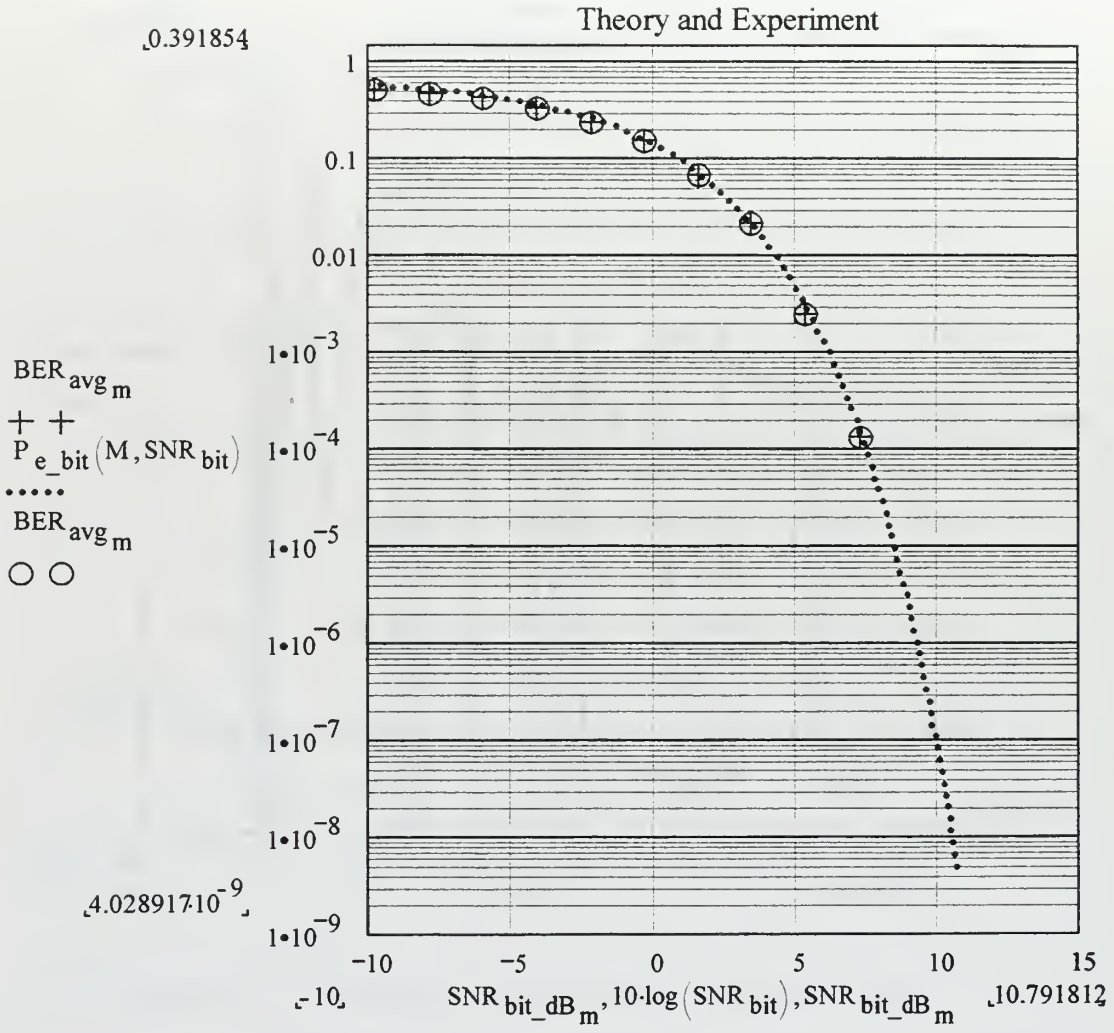

Figure 25. Probability of Bit Error (Theory and Simulation) for Coherent 8FSK Detection.

The relative difference between the theoretical probability of error for coherent 8FSK and the probability of error estimates obtained by the simulation is shown in Figure 26.

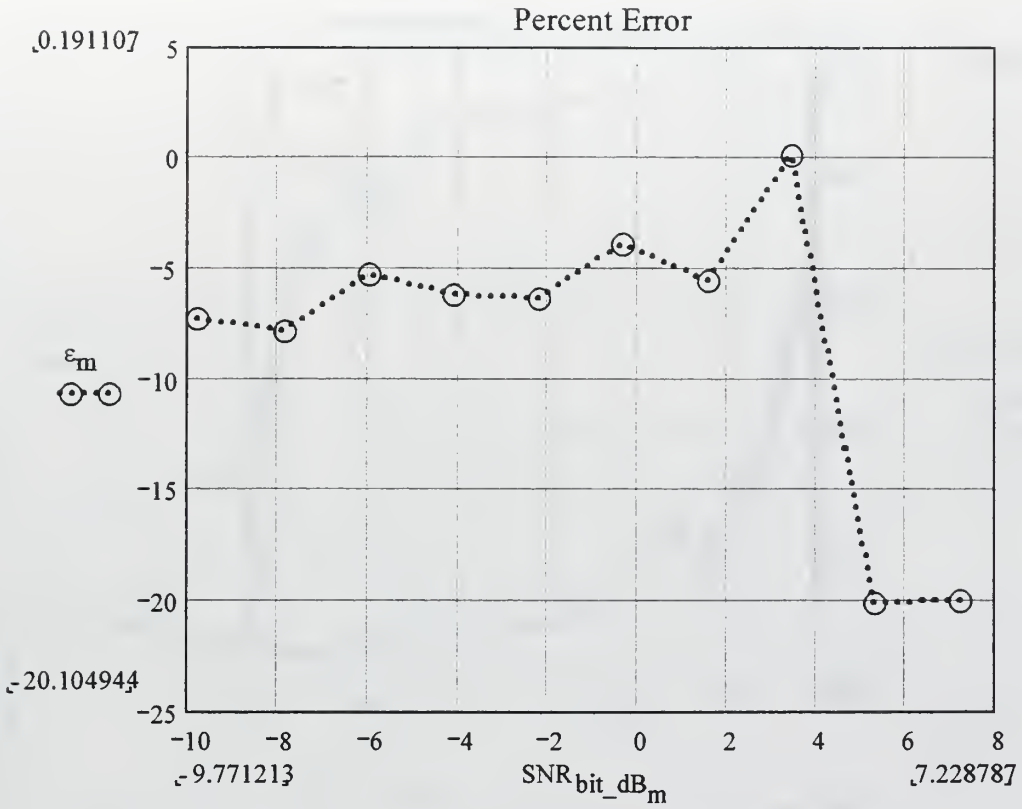

Figure 26. Percent difference between Theoretical and Simulation Results for Coherent Detection of 8FSK.

The convergence of the estimate the bit error probability as a function of the number of symbols transmitted in the course of the simulation is shown in Figure 27. This allows us to estimate the minimum number of symbols required for a particular simulation and to verify that the minimum has been reached. For this particular simulation, the convergence is evident at approximately $2.5 \cdot 10^4$ symbols.

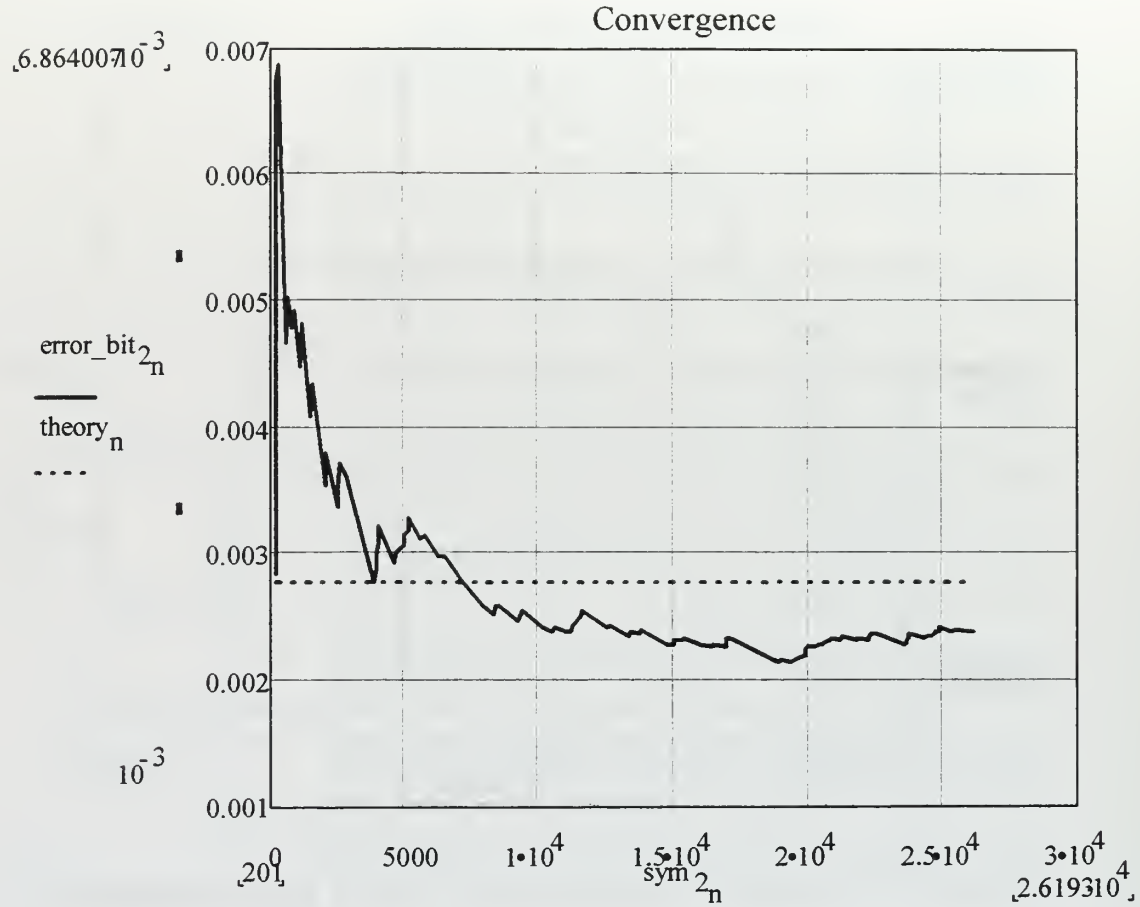

Figure 27. Convergence Plot for Coherent Detection of 8FSK.

The accuracy of the simulation estimate for the bit error probability for coherent 8FSK is shown in Figure 28. The 'oscillatory' convergence to approximately zero as the number of symbols increases is evident.

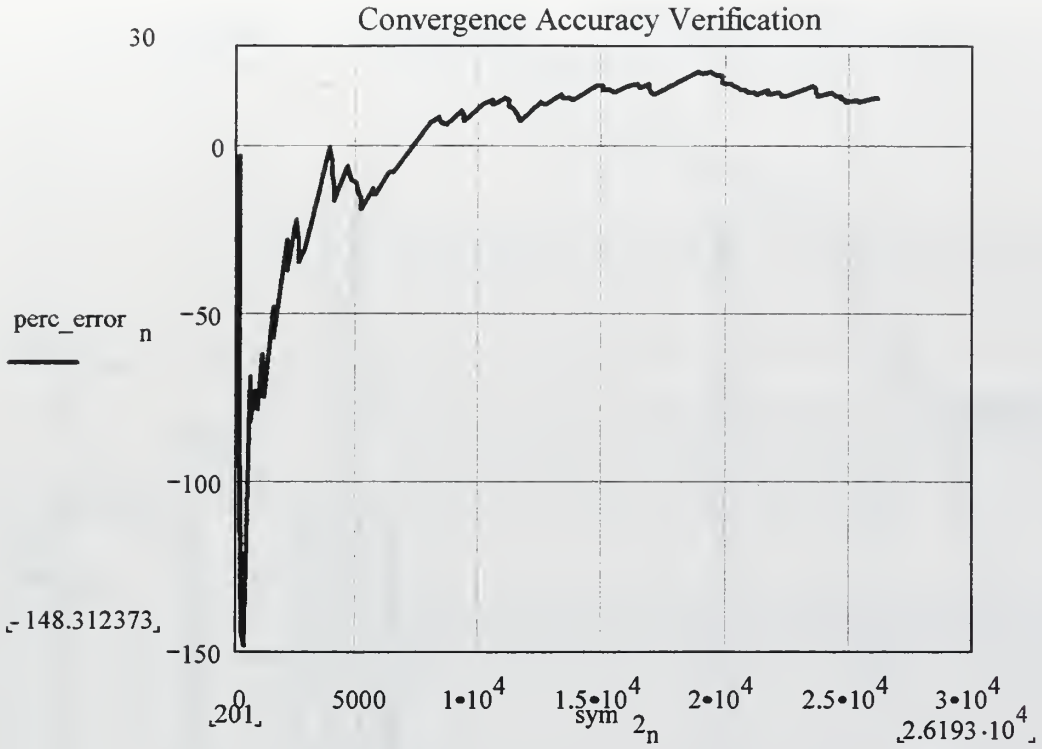

Figure 28. Percent Difference Between Theory and Simulation for Coherent 8FSK as a function of the Number of Symbols.

The differences between the theoretical bit error probabilities for coherent 8FSK and the simulation estimates for various SNRs are plotted in Figure 29 as a histogram. All estimates are within the -22% to +0% interval about the theoretical values, with most differences between -7% and -3%.

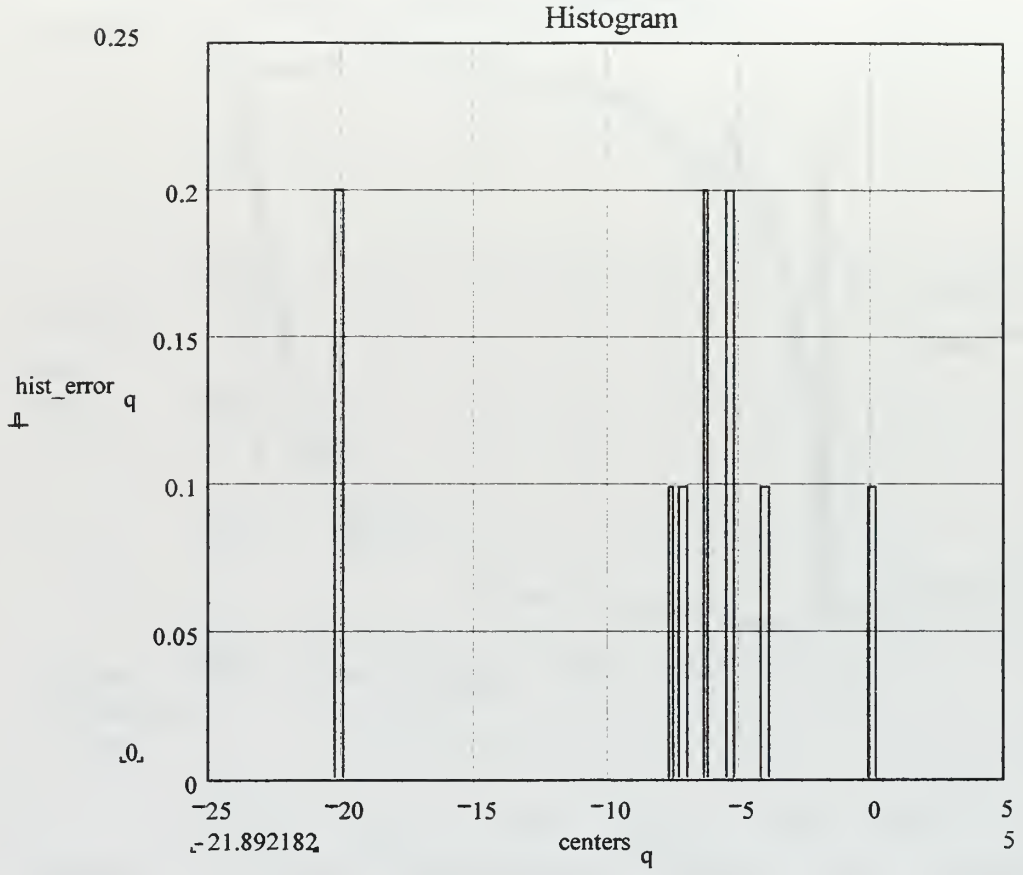

Figure 29. Histogram of Simulation Errors for Coherent 8FSK.

4. Results For Noncoherent BFSK

The theoretical and the experimental bit error probability for noncoherently detected binary FSK (BFSK or 2FSK) as a function of the SNR in dB is shown in Figure 30.

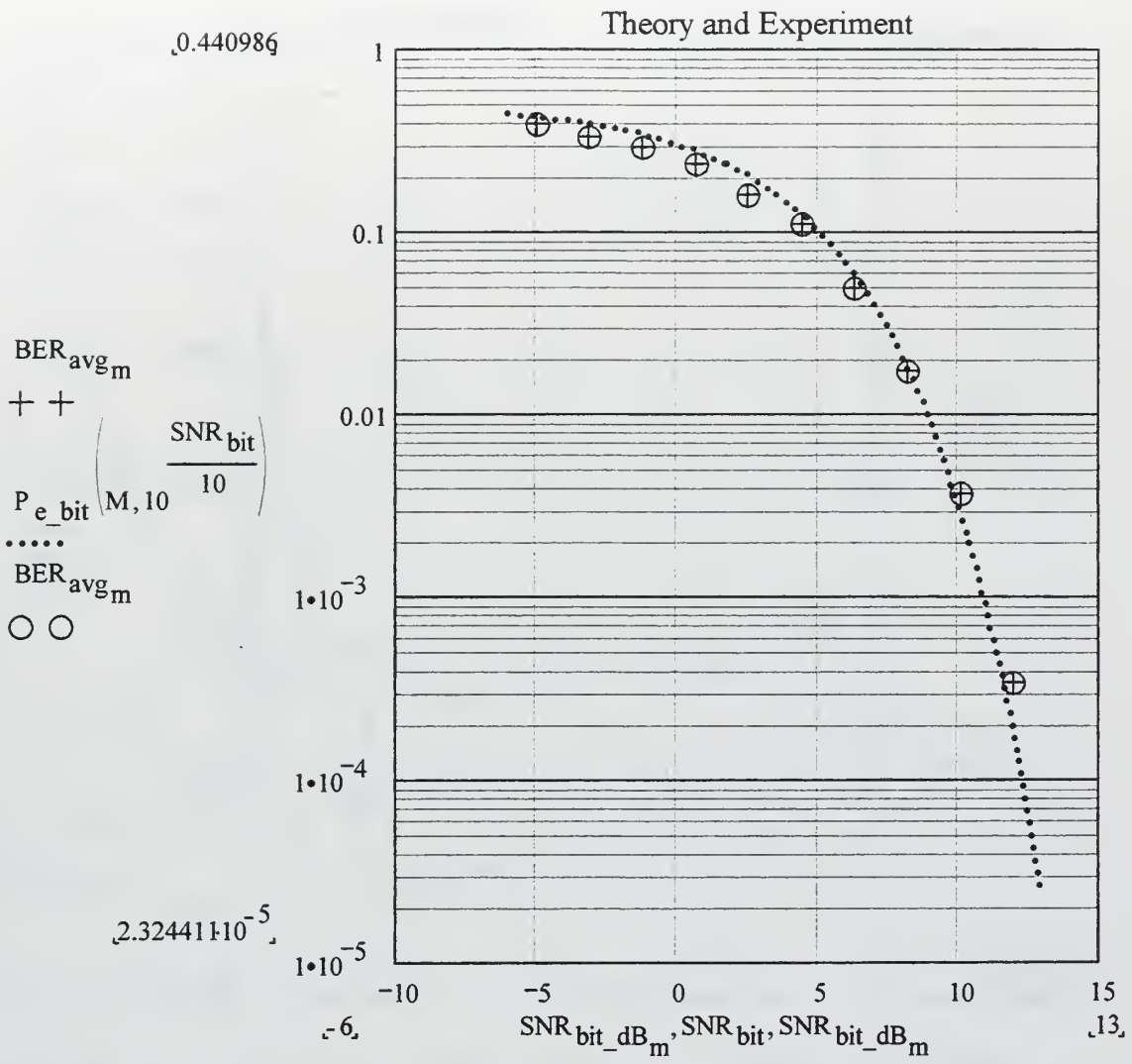

Figure 30. Probability of Bit Error (Theory and Simulation) for Noncoherent BFSK Detection.

The relative difference between the theoretical probability of error for noncoherent BFSK and the probability of error estimates obtained by the simulation is shown in Figure 31.

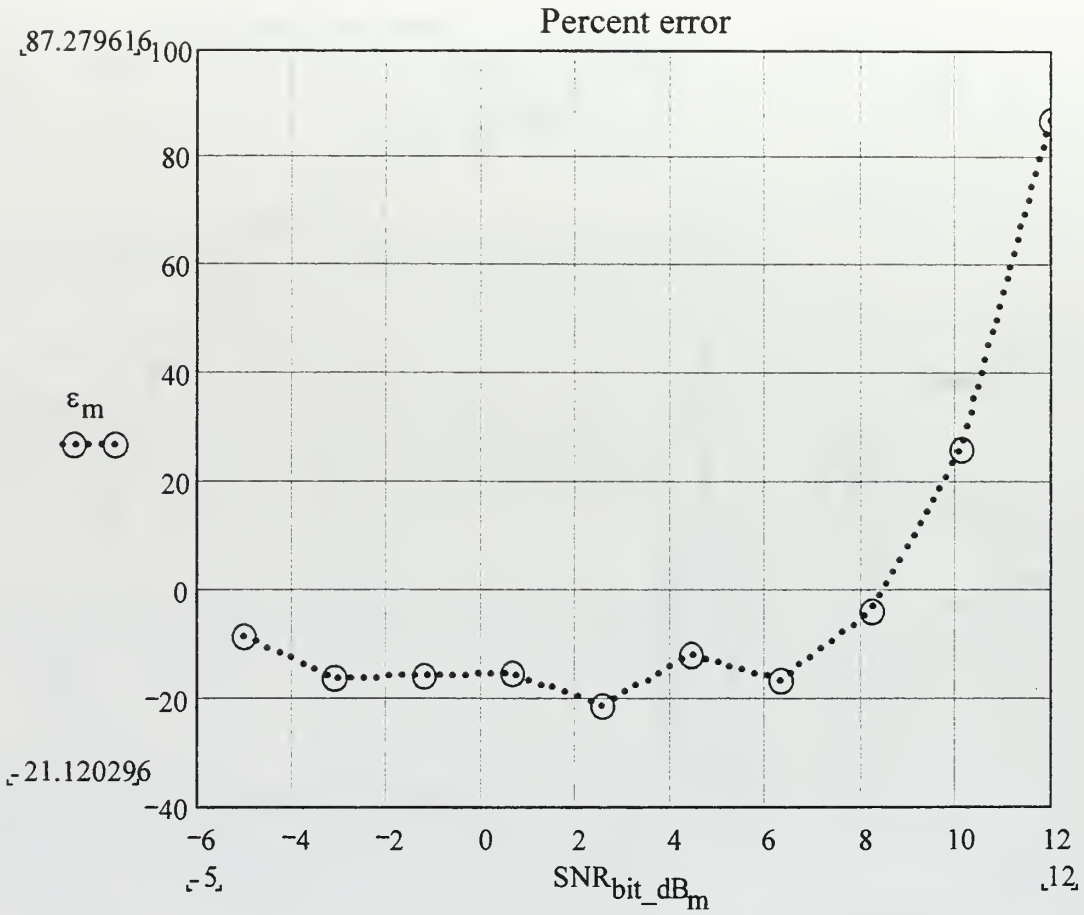

Figure 31. Percent difference between Theoretical and Simulation Results for Noncoherent Detection of BFSK.

The convergence of the estimate of the bit error probability as a function of the number of symbols transmitted in the course of the simulation is shown in Figure 32. This allows us to estimate the minimum number of symbols required for a particular simulation and to verify that the minimum has been reached. For this particular

simulation, the convergence is evident at approximately $2.3 \cdot 10^4$ symbols.

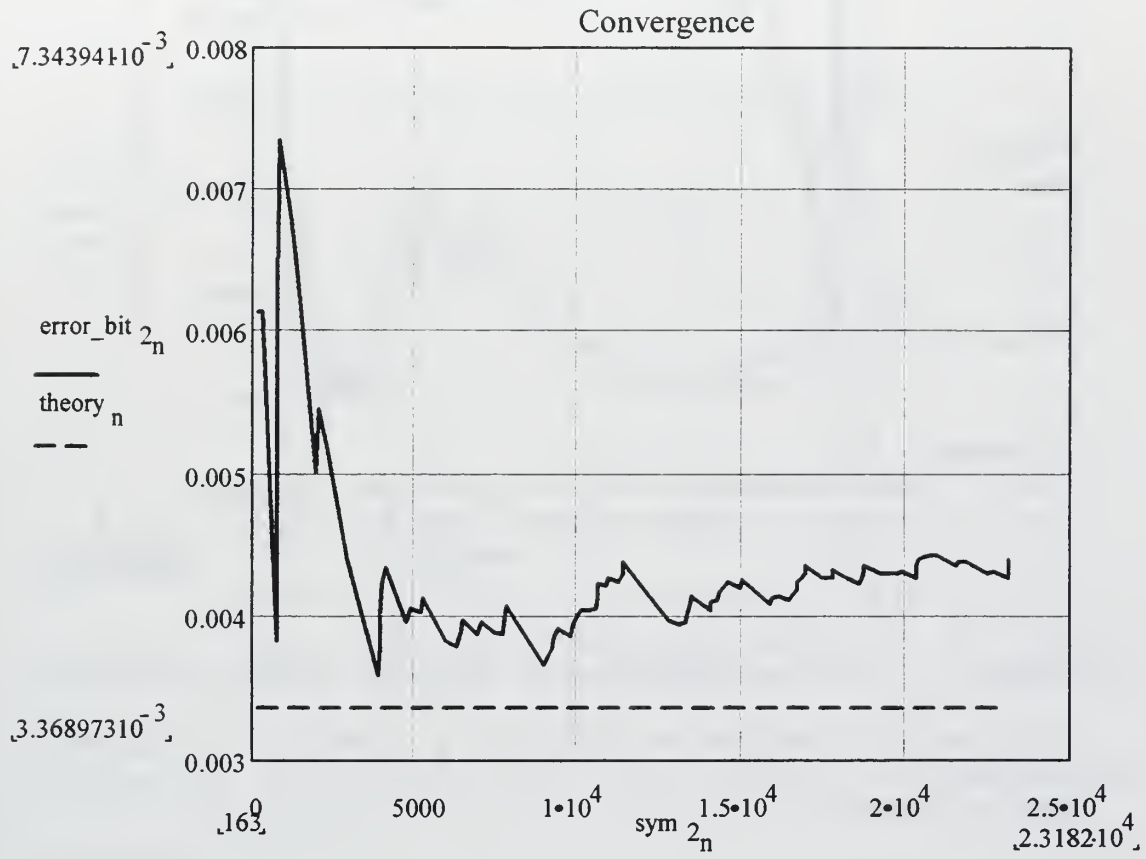

Figure 32. Convergence Plot (100 errors) for Noncoherent Detection of BFSK.

The accuracy of the simulation estimate for the bit error probability for noncoherent BFSK is shown in Figure 33. In this case, the convergence is to a non-zero value (approximately -30% off) as the number of symbols increases.

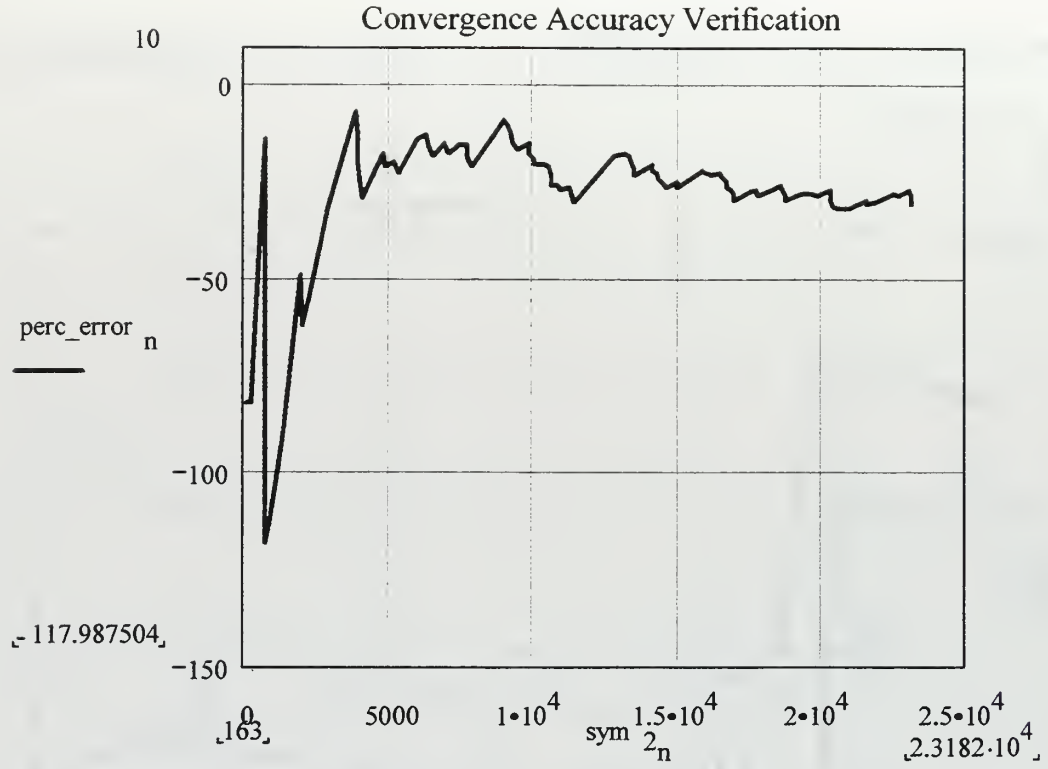

Figure 33. Percent Difference Between Theory and Simulation for Noncoherent BFSK as a function of the Number of Symbols.

The convergence of the estimate of the bit error probability as a function of the number of symbols transmitted in the course of the simulation using 200 minimum acceptable errors is shown in Figure 34. This allows us to estimate the minimum number of symbols required for a particular simulation and to verify that the minimum has been reached. For this particular simulation, the convergence is evident at approximately $8 \cdot 10^4$ symbols.

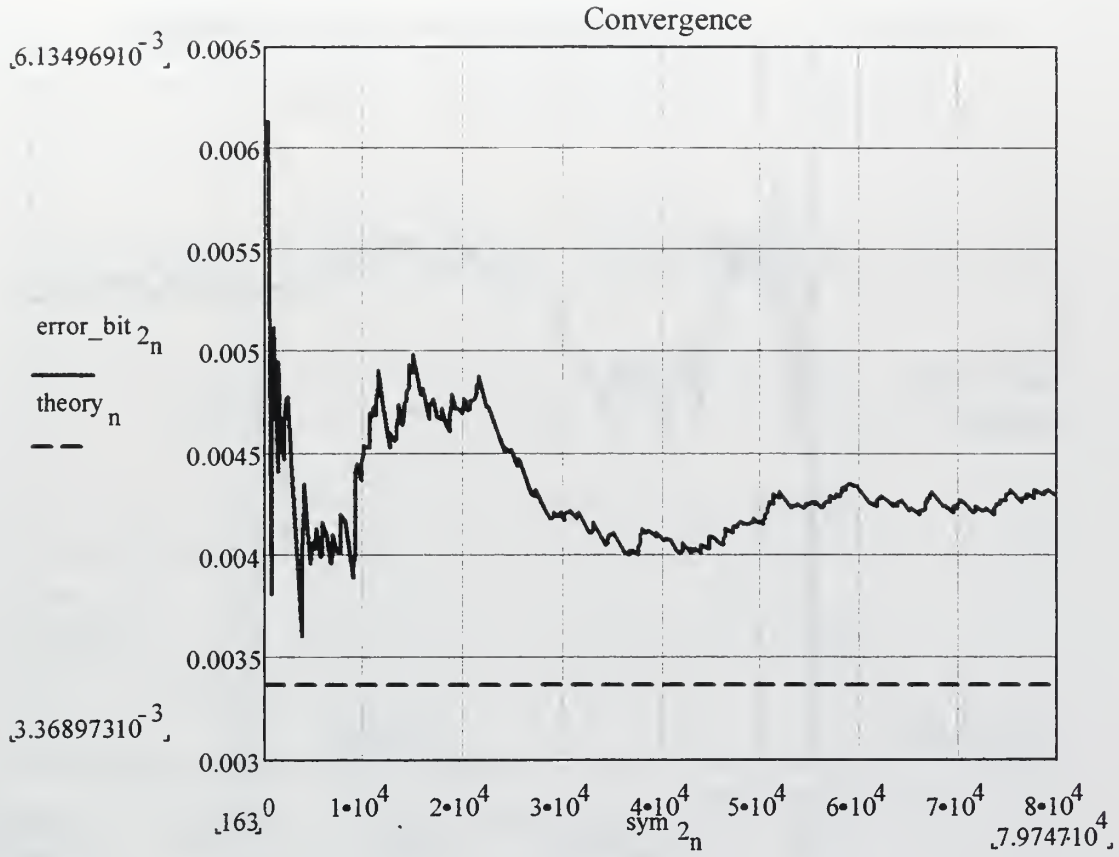

Figure 34. Convergence Plot (200 errors) for Noncoherent Detection of BFSK.

The accuracy of the simulation estimate for the bit error probability for noncoherent BFSK is shown in Figure 35. In this case, the convergence is again to a non-zero value (approximately -30% off) as the number of symbols increases.

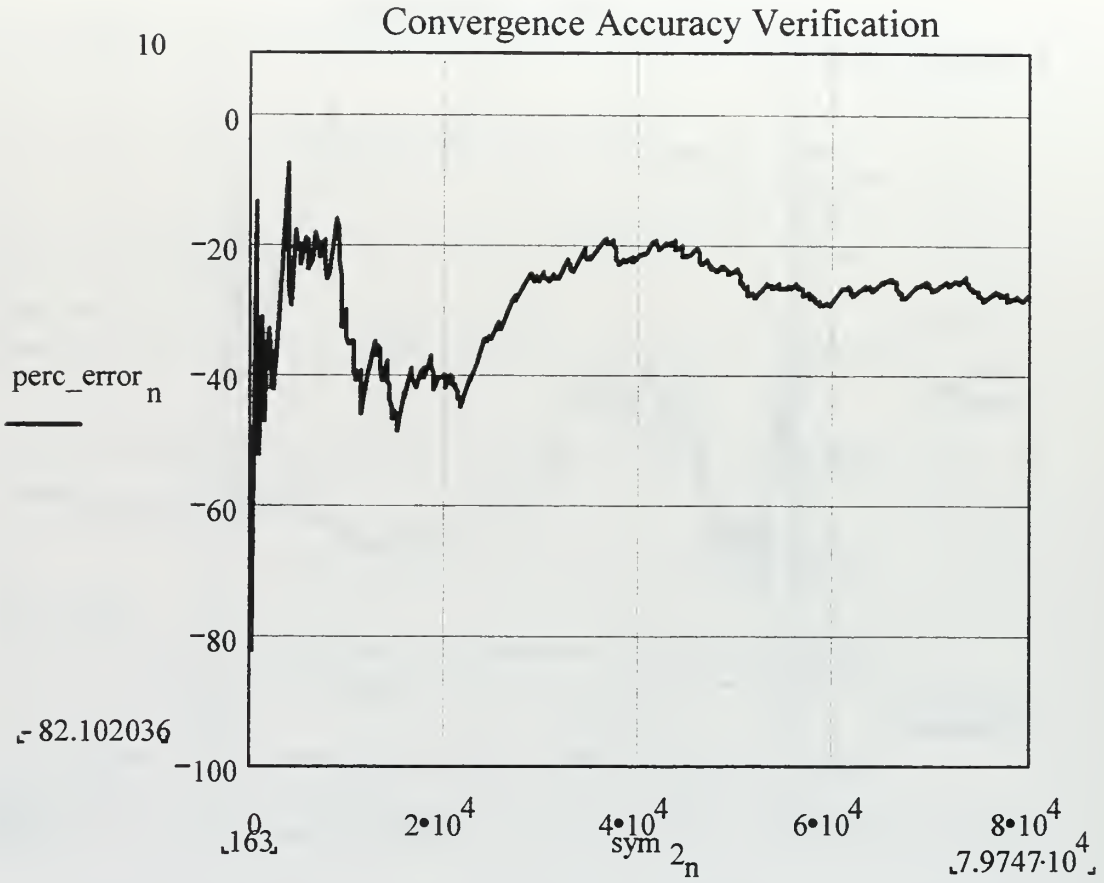

Figure 35. Percent Difference Between Theory and Simulation for Noncoherent BFSK (200 Errors) as a function of the Number of Symbols.

The convergence of the estimate of the bit error probability as a function of the number of symbols transmitted in the course of the simulation using 300 minimum acceptable errors is shown in Figure 36. This allows us to estimate the minimum number of symbols required for a particular simulation, and to verify that the minimum has

been reached. For this particular simulation, the convergence is evident at approximately $9 \cdot 10^4$ symbols.

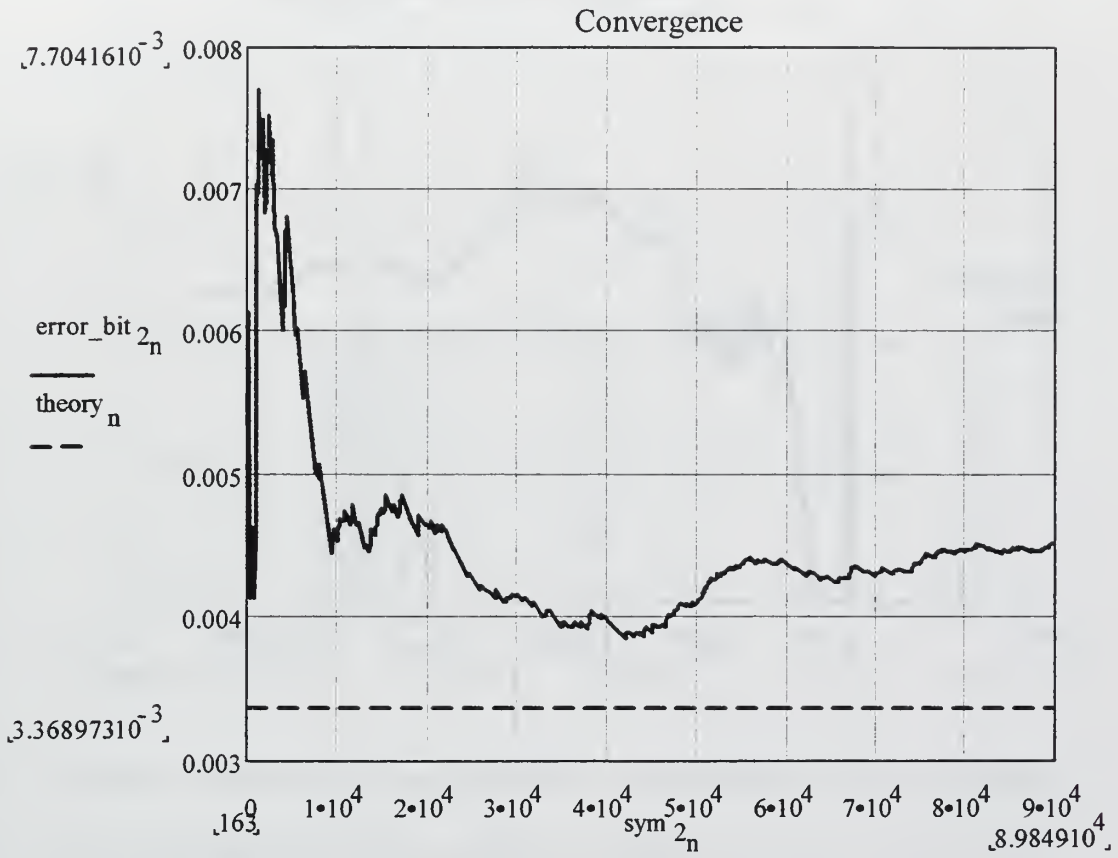

Figure 36. Convergence Plot (300 errors) for Noncoherent Detection of BFSK.

The accuracy of the simulation estimate for the bit error probability for noncoherent BFSK is shown in Figure 37. In this case, the convergence is again to a non-zero

value (approximately -30% off) as the number of symbols increases.

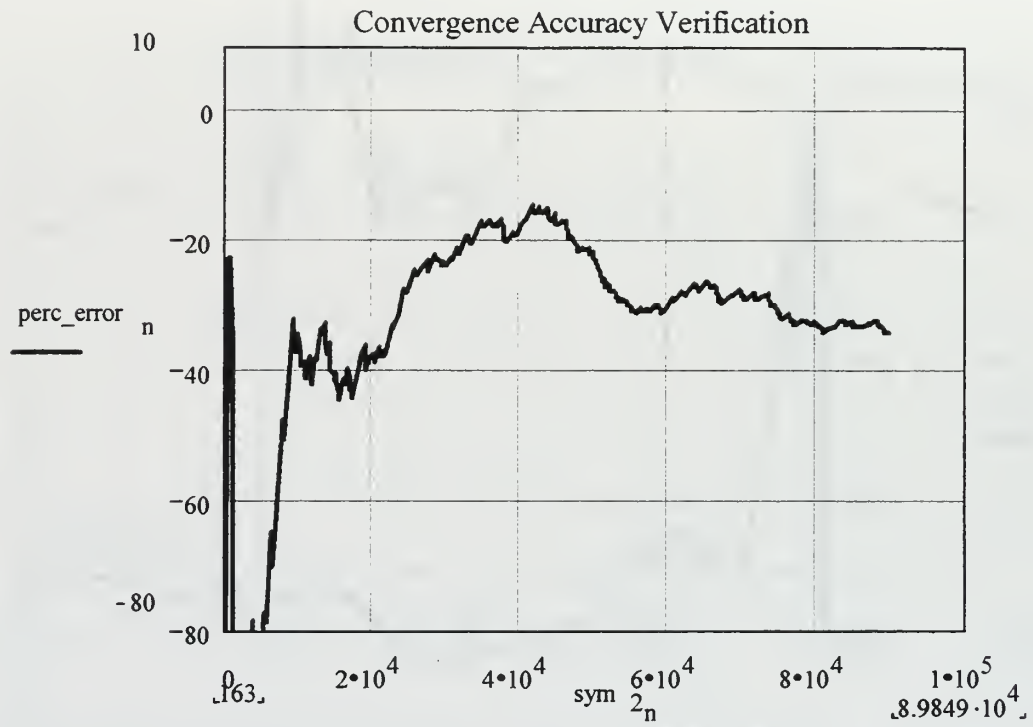

Figure 37. Percent Difference Between Theory and Simulation for Noncoherent BFSK (300 Errors) as a function of the Number of Symbols.

The differences between the theoretical bit error probabilities for noncoherent BFSK (100 errors case) and the simulation estimates for various SNRs are plotted in Figure 38 as a histogram. All estimates are within the -22% to +87% interval about the theoretical values, with most differences between -22% and 0%.

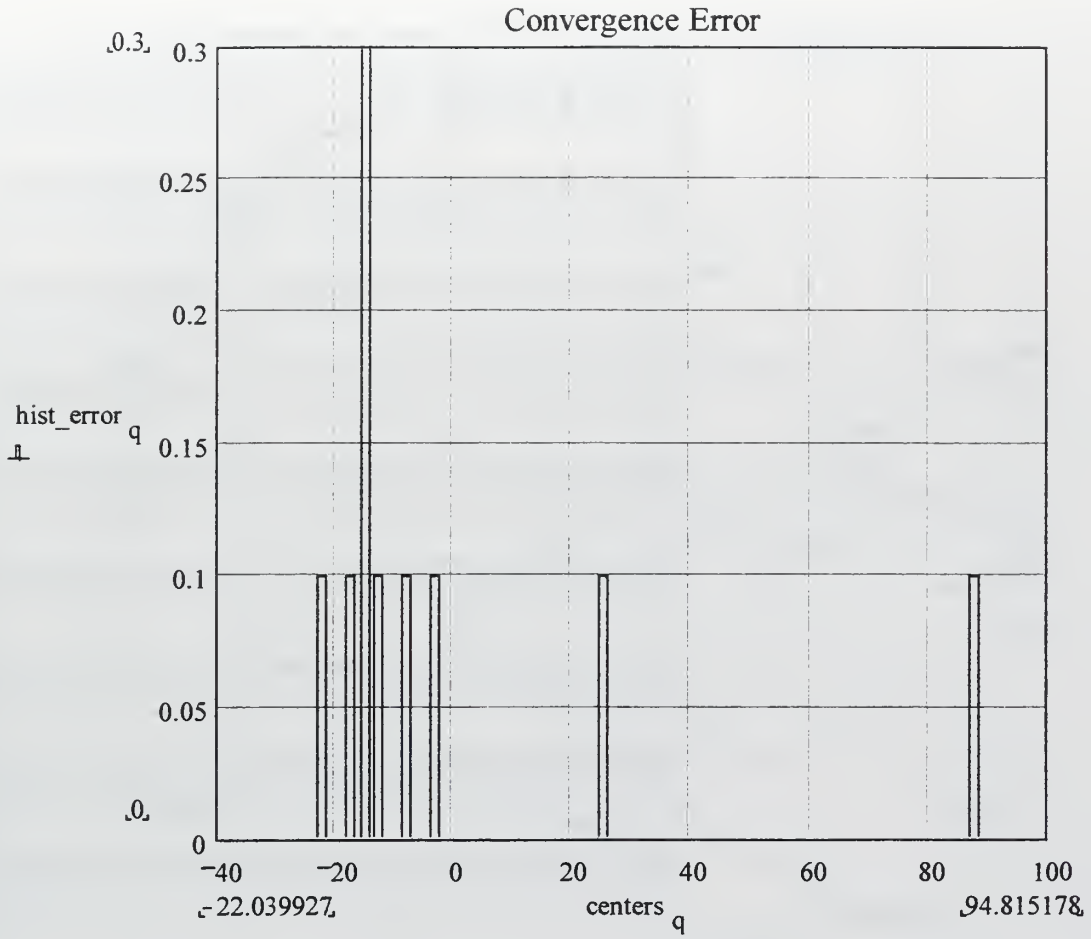

Figure 38. Histogram of Simulation Errors (100) for Noncoherent BFSK Detection.

5. Results For Noncoherent 4FSK

The theoretical and the experimental bit error probability for noncoherently detected 4FSK as a function of the SNR in dB is shown in Figure 39.

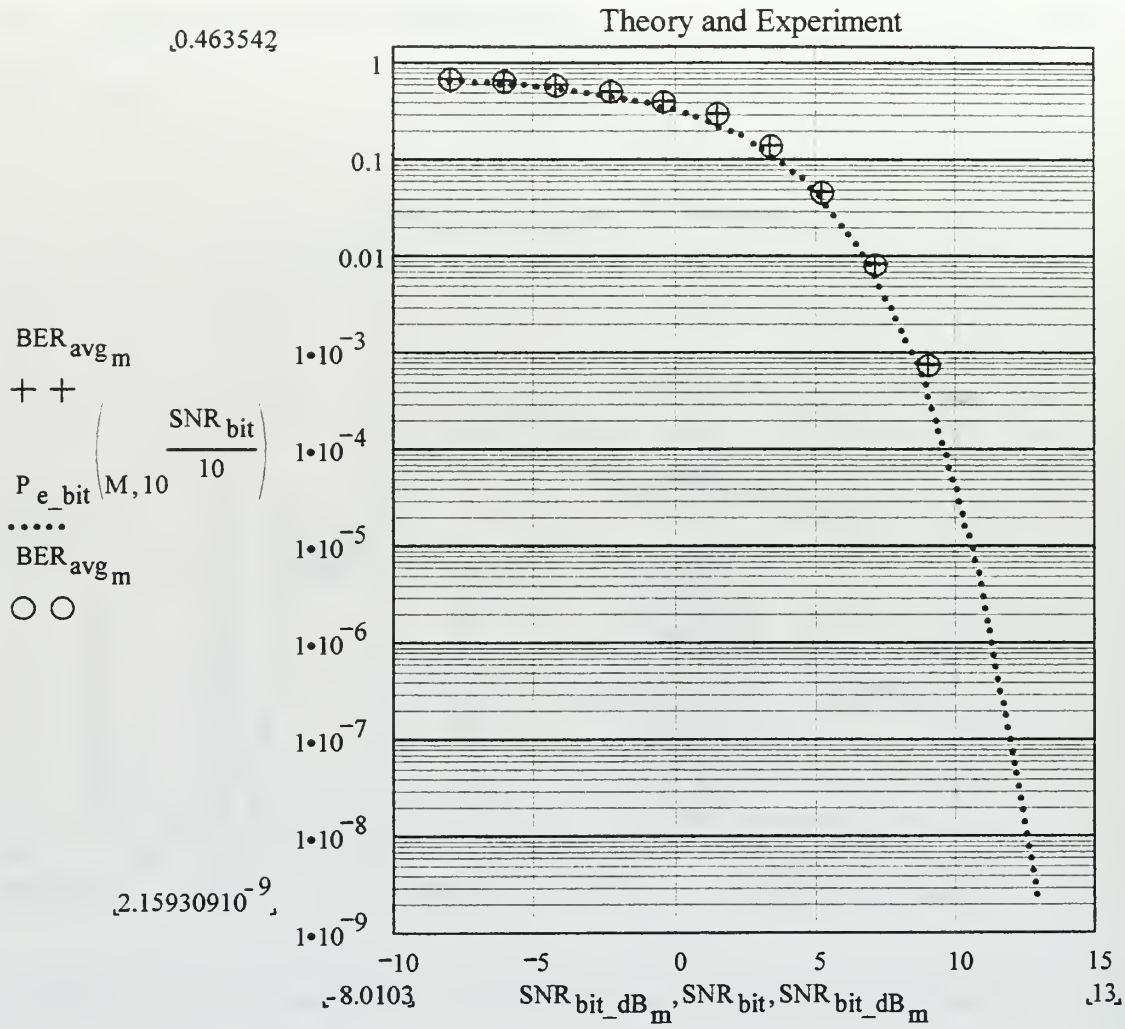

Figure 39. Probability of Bit Error (Theory and Simulation) for Noncoherent 4FSK Detection.

The relative difference between the theoretical probability of error for noncoherent 4FSK and the probability of error estimates obtained by the simulation is shown in Figure 40.

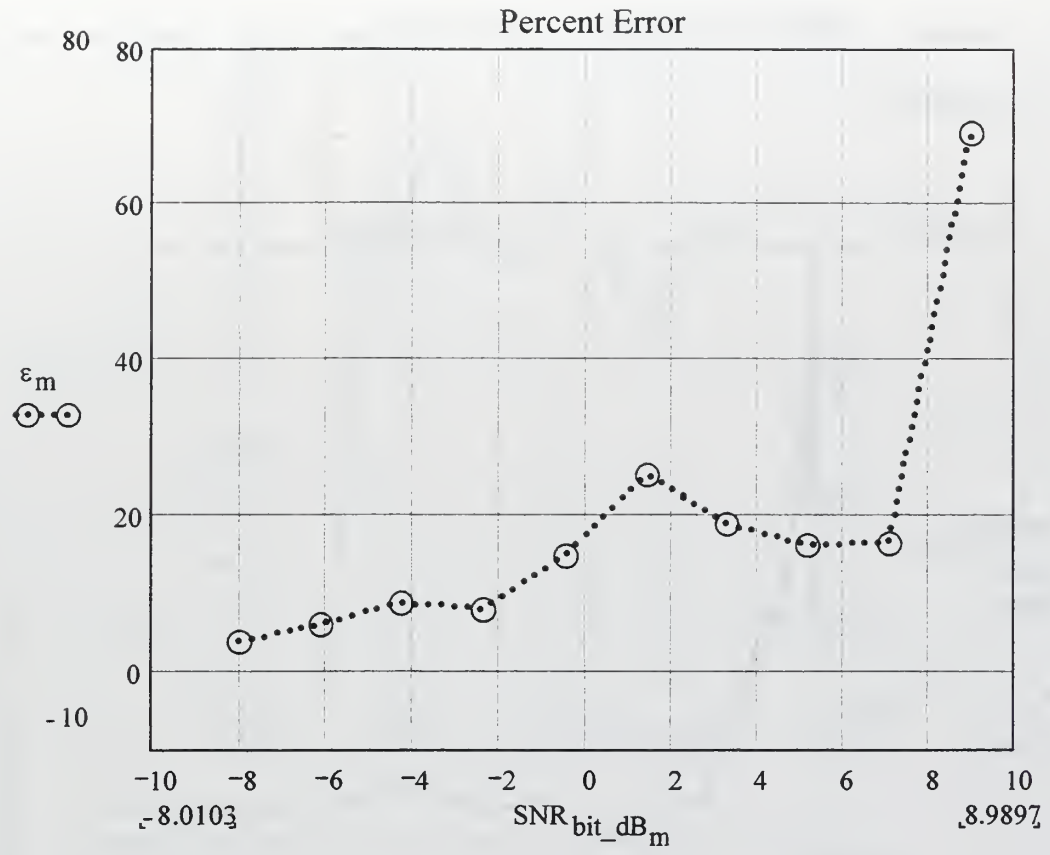

Figure 40. Percent difference between Theoretical and Simulation Results for Noncoherent Detection of 4FSK.

The convergence of the estimate of the bit error probability as a function of the number of symbols transmitted in the course of the simulation is shown in Figure 41. This allows us to estimate the minimum number of symbols required for a particular simulation and to verify that the minimum has been reached. For this particular

simulation, the convergence is evident at approximately $9 \cdot 10^3$ symbols.

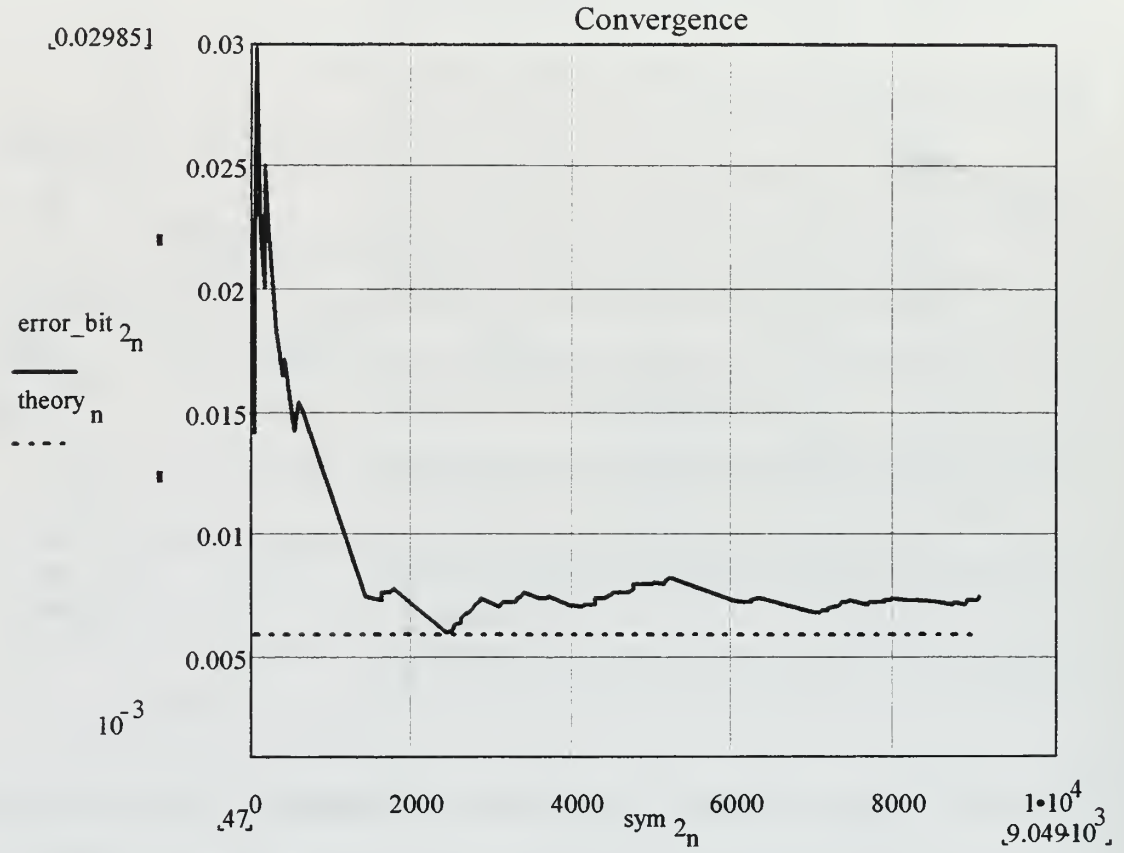

Figure 41. Convergence Plot for Noncoherent Detection of 4FSK.

The accuracy of the simulation estimate for the bit error probability for noncoherent 4FSK is shown in Figure 42. In this case the convergence is again to a non-zero value (approximately -20% off) as the number of symbols increases.

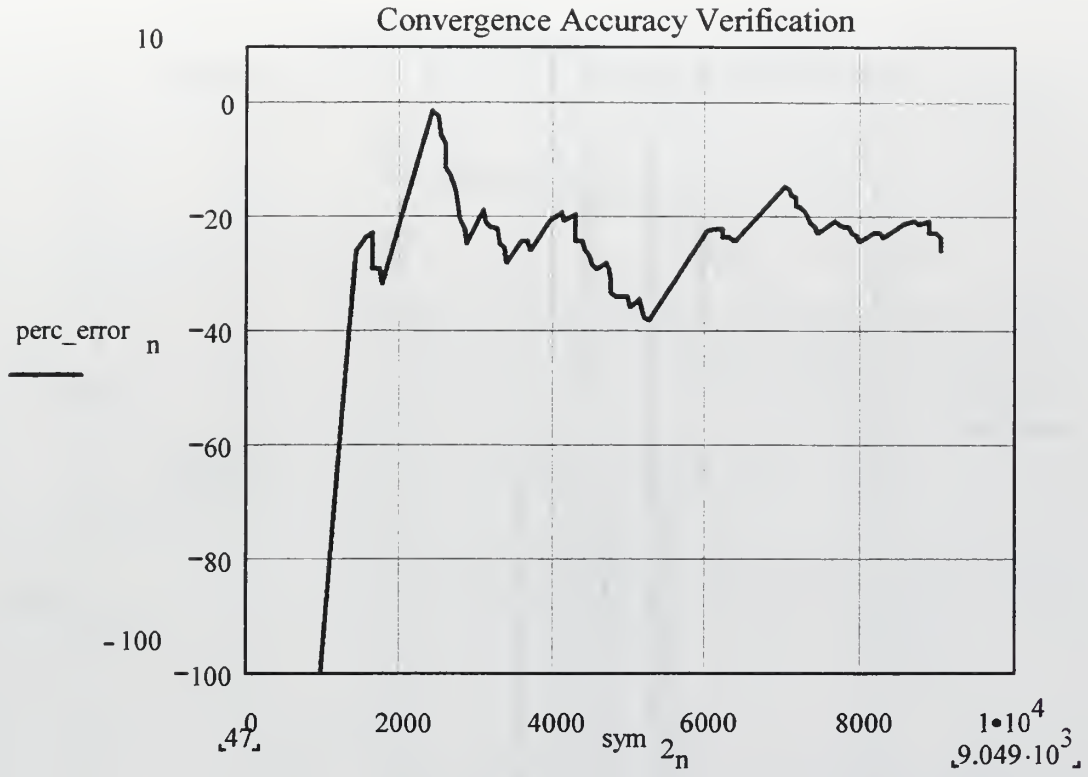

Figure 42. Percent Difference Between Theory and Simulation for Noncoherent 4FSK as a function of the Number of Symbols.

The differences between the theoretical bit error probabilities for coherent 4FSK and the simulation estimates for various SNRs are plotted in Figure 43 as a histogram. All estimates are within the 5% to 70% interval about the theoretical values, with most differences between 5% and 20%.

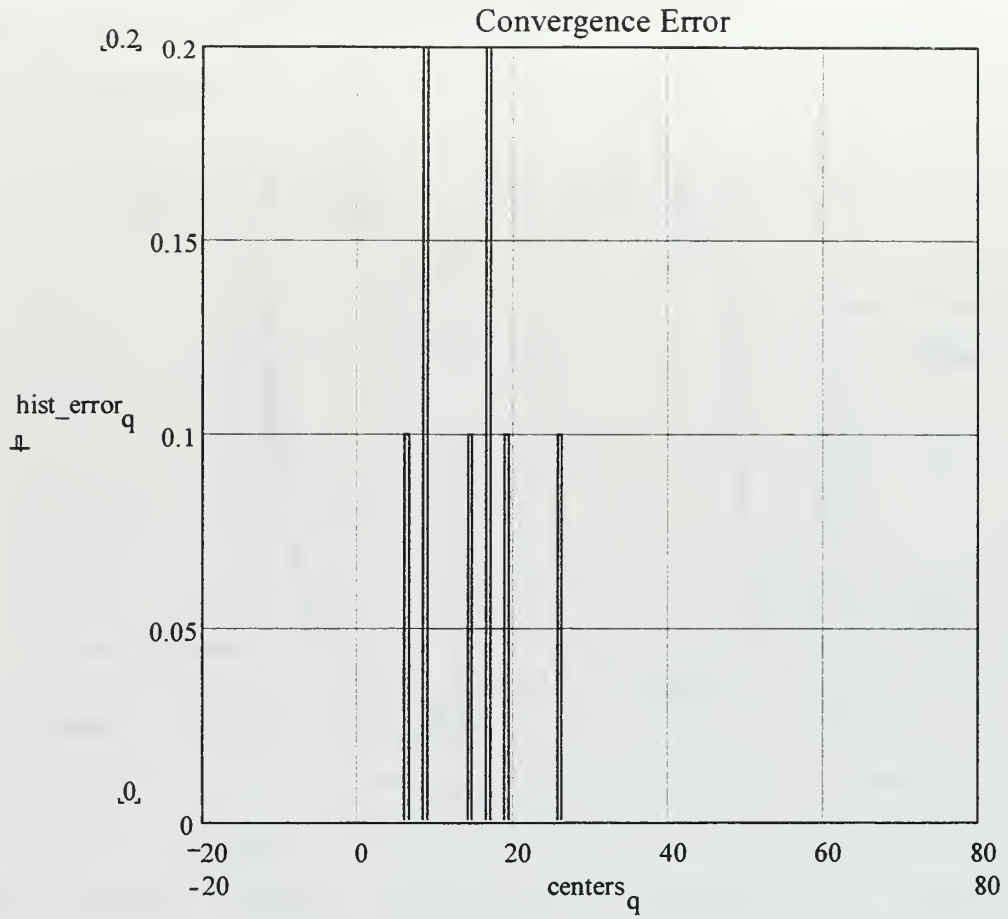

Figure 43. Histogram of Simulation Errors for Noncoherent 4FSK.

6. Results For Noncoherent 8FSK

The theoretical and the experimental bit error probability for noncoherently detected 8FSK as a function of the SNR in dB is shown in Figure 44.

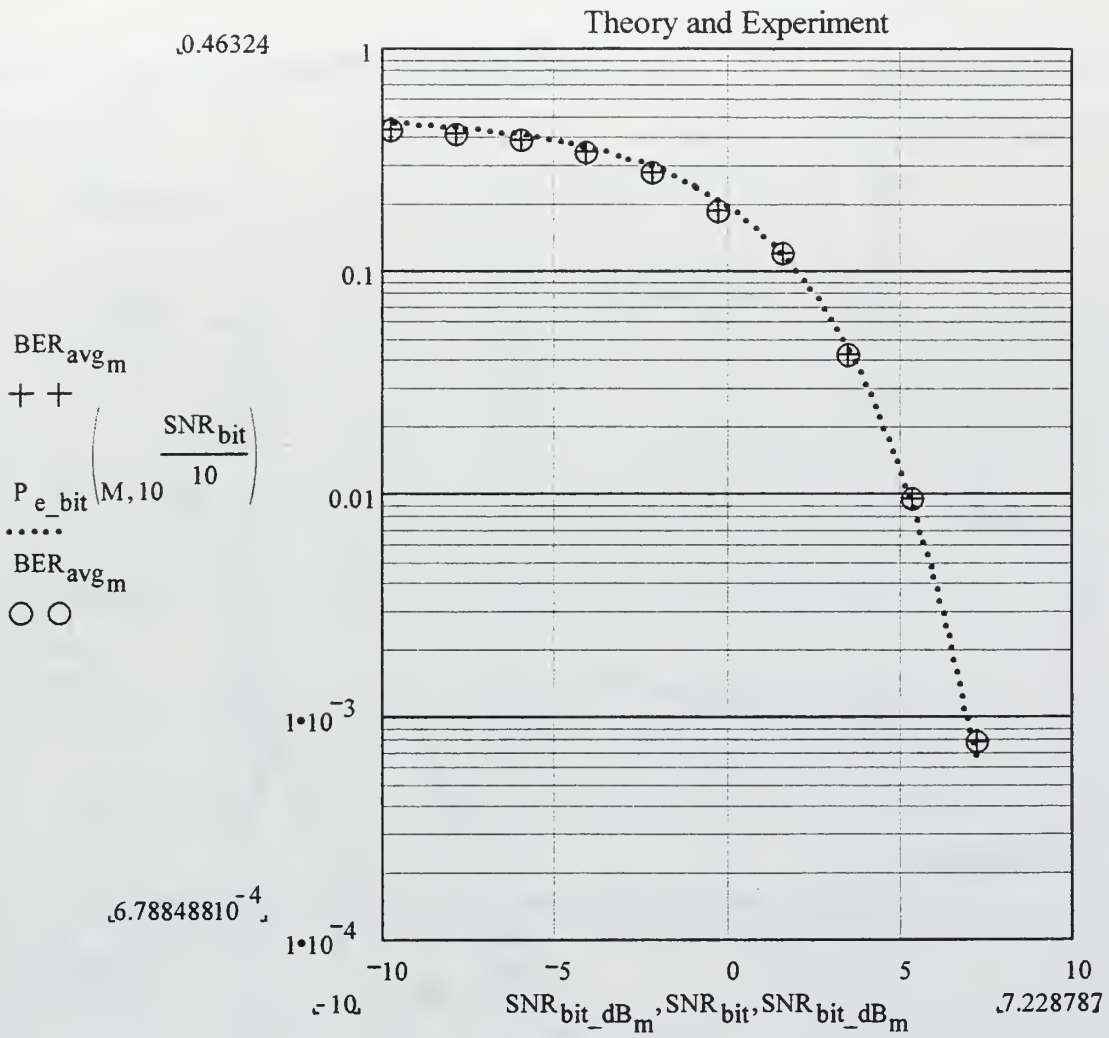

Figure 44. Probability of Bit Error (Theory and Simulation) for Noncoherent 8FSK Detection.

The relative difference between the theoretical probability of error for noncoherent 8FSK and the probability of error estimates obtained by the simulation is shown in Figure 45.

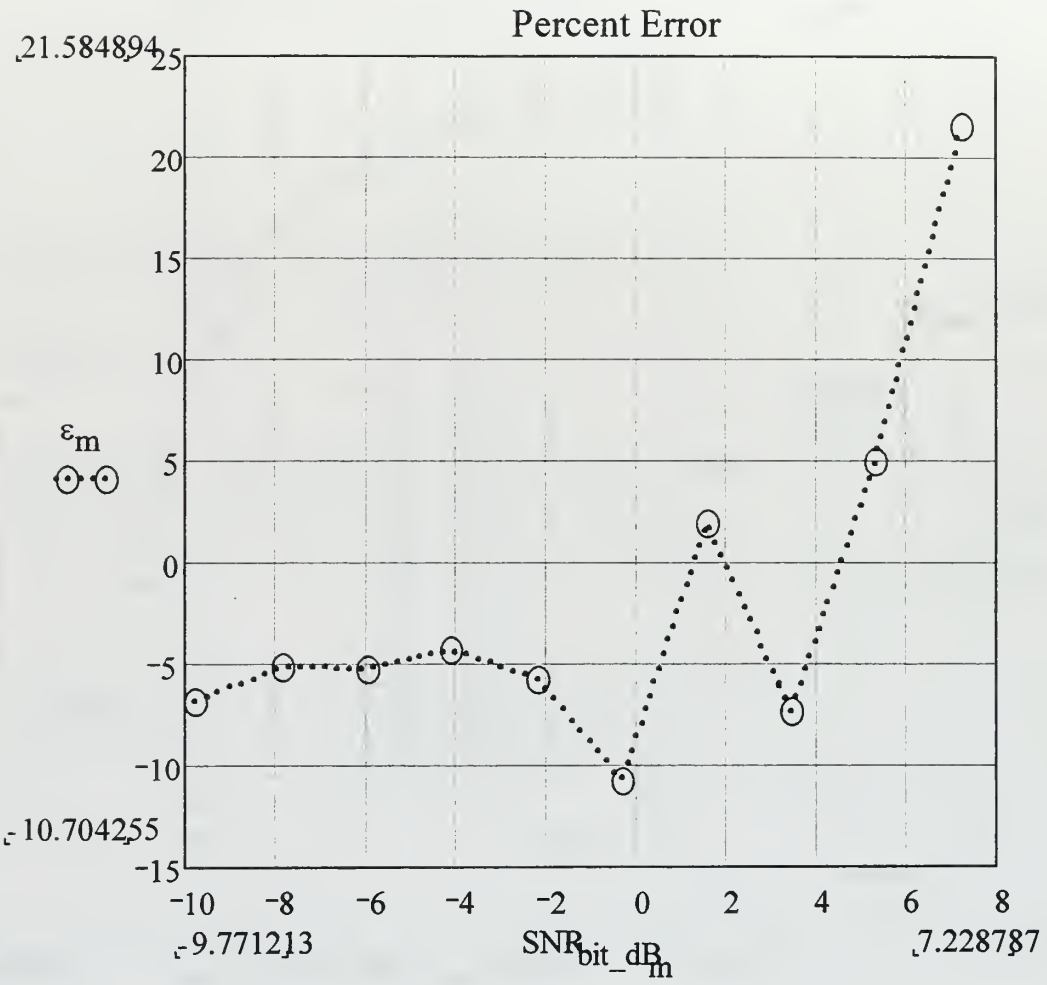

Figure 45. Percent difference between Theoretical and Simulation Results for Noncoherent Detection of 8FSK.

The convergence of the estimate of the bit error probability as a function of the number of symbols transmitted in the course of the simulation is shown in Figure 46. This allows us to estimate the minimum number of

symbols required for a particular simulation and to verify that the minimum has been reached. For this particular simulation, the convergence is evident for approximately $5 \cdot 10^3$ symbols.

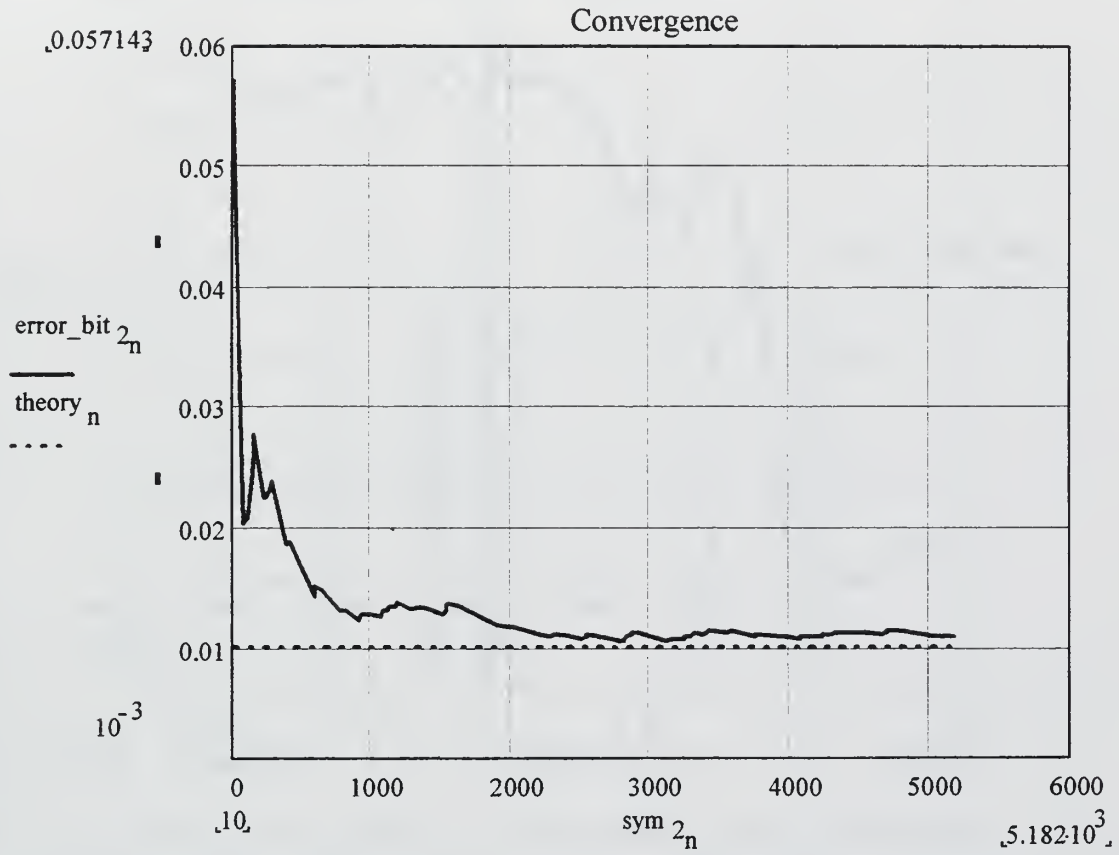

Figure 46. Convergence Plot for Noncoherent Detection of 8FSK.

The accuracy of the simulation estimate for the bit error probability for noncoherent 8FSK is shown in Figure 47. In

this case, the convergence is again to a non-zero value (approximately -13% off) as the number of symbols increases.

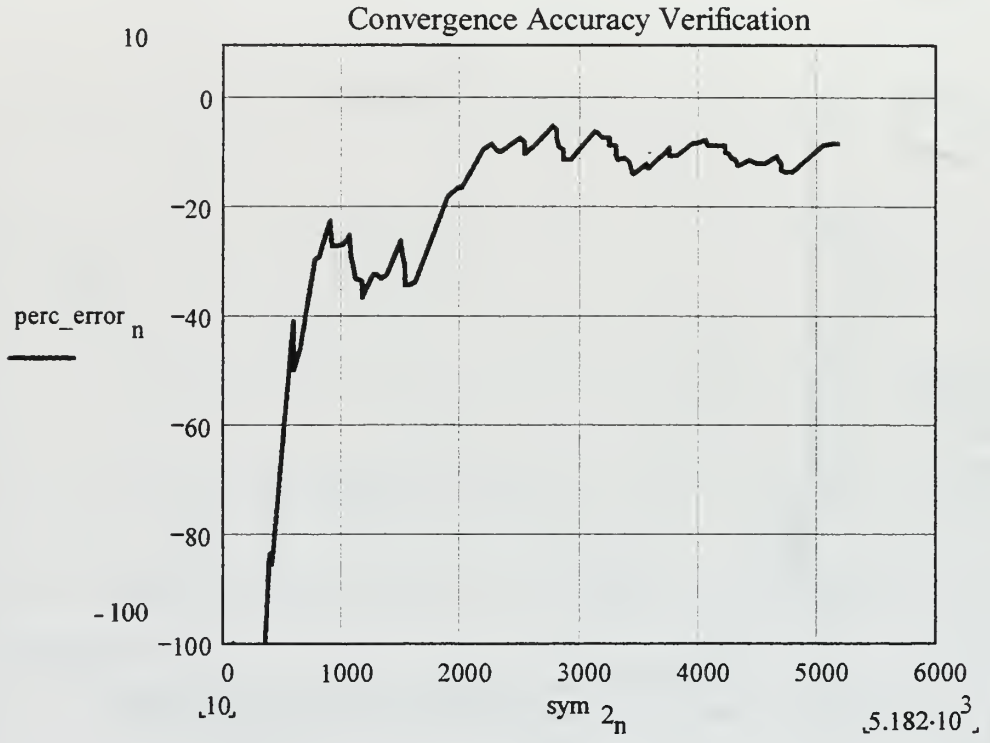

Figure 47. Percent Difference Between Theory and Simulation for Noncoherent 8FSK as a function of the Number of Symbols.

The differences between the theoretical bit error probabilities for coherent 4FSK and the simulation estimates for various SNR's are plotted in Figure 48 as a histogram. All estimates are within the -12% to +5% interval about the

theoretical values, with most differences between -7% and -3%.

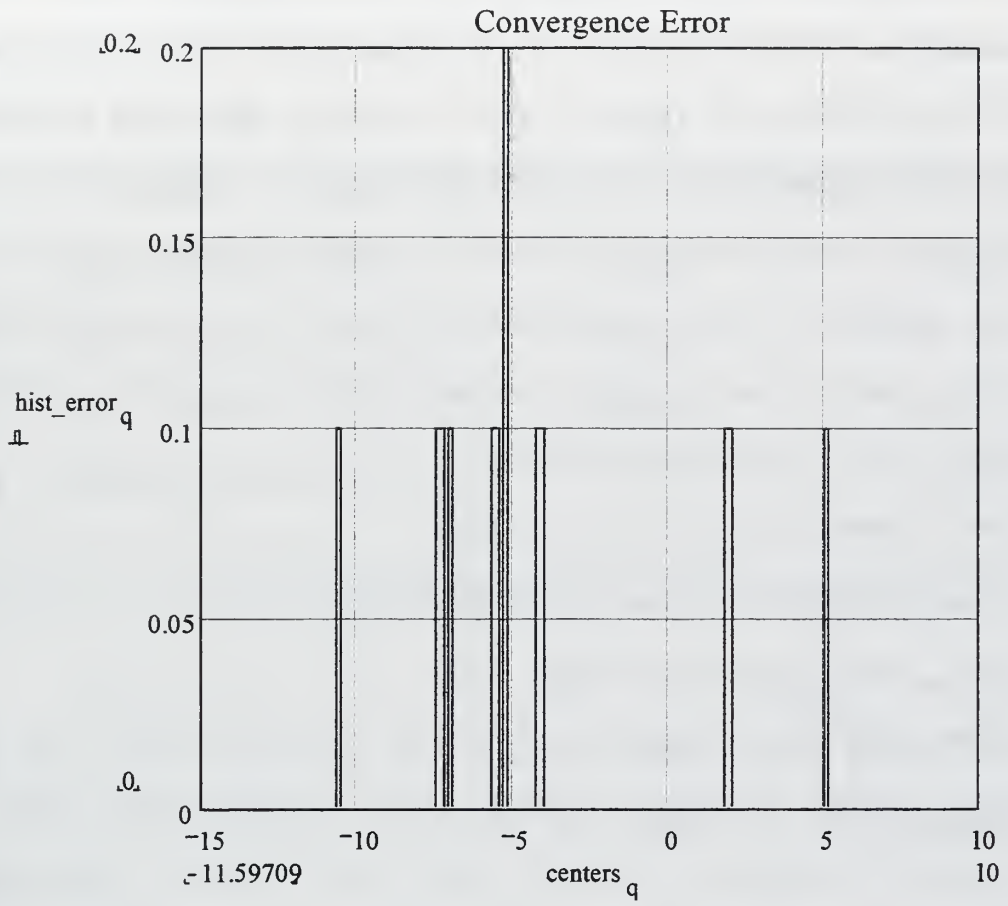

Figure 48. Histogram of Simulation Errors for Noncoherent 8FSK.

B. OBSERVATIONS

Simulation results for coherent MFSK in the presence of AWGN agree very well with theory. As can be seen from the convergence plots, in all cases of coherent MFSK detection

the differences between the bit error probability estimates and the theoretical values converge to zero as the number of symbols increases, which means that the simulation estimates are correct. However, this is not the case for noncoherent MFSK. The simulation results are too small for BFSK and 8FSK (bit error probability is underestimated) compared to the theoretical ones. Only in case of 4FSK are the simulation results higher (bit error probability is overestimated) than the theoretical ones. The simulation results converge to values that are approximately:

- 30% lower for noncoherent BFSK
- 20% higher for noncoherent 4FSK and
- 13% lower for noncoherent 8FSK.

The simulation results for both noncoherent 4FSK and 8FSK are within expected bounds. Only noncoherent BFSK is not within expected bounds for the 95.4% confidence intervals for 100 errors of Monte Carlo Simulations. An increase in the observed number of errors to 200 and 300 does not alter the simulation results, which still converge to the same somewhat incorrect values. This can be seen in Figure 33, 34, 35, and 36 for noncoherent BFSK where, despite the fact that 100, 200, and 300 errors were observed, the bit error probability still converged to about

the same (incorrect) value. This suggests a systematic error in the Communications Toolbox for noncoherent BFSK.

The summary of the accuracy of the simulation for both coherent and noncoherent MFSK detection is presented in Tables 4,5, and 6.

	Bit Error Probability	Coherent BFSK	NoncoherentBFSK
1.	Mean Percent Error	-2.532	0.538
2.	Max. Percent Error	9.065	87.28
3.	Min. Percent Error	-18.209	-21.12
4.	Standard Deviation	7.777	31.571

Table 4. Comparison of the Simulation and the Theoretical Results for Coherent and Noncoherent BFSK Detection.

From Table 4, we note that the standard deviation is much larger for the noncoherent MFSK. This means that the simulation results are less accurate for noncoherent BFSK detection.

	Bit Error Probability	Coherent 4FSK	NonCoherent 4FSK
1.	Mean Percent Error	13.628	18.946
2.	Max. Percent Error	38.725	4.101
3.	Min. Percent Error	-10.883	69.43
4.	Standard Deviation	14.109	17.931

Table 5. Comparison of the Simulation and the Theoretical Results for Coherent and NonCoherent 4FSK Detection.

From Table 5, we note that the mean values and the standard deviations for coherent and noncoherent 4FSK are closer to each other (the mean values in the absolute sense), so the accuracy of the simulation is comparable for both coherent and noncoherent 4FSK.

	Bit Error Probability	Coherent 8FSK	Noncoherent 8FSK
1.	Mean Percent Error	-8.207	-1.666
2.	Max. Percent Error	0.191	21.585
3.	Min. Percent Error	-20.105	-10.704
4.	Standard Deviation	6.281	8.873

Table 6. Comparison of the Simulation and the Theoretical Results for Coherent and Noncoherent 8FSK Detection.

From Table 6, we note that the mean values and the standard deviations for coherent and noncoherent 8FSK are close to each other (the mean values in the absolute sense), so the accuracy of the simulation is in general comparable for both coherent and noncoherent 8FSK. It is interesting to note that as the number of symbols M increases from 2 to 8 the simulation results for noncoherent MFSK become more accurate while the simulation results for coherent MFSK become less accurate.

V. SIMULATION AND PERFORMANCE ANALYSIS FOR CO-CHANNEL INTERFERENCE

A. INTERFERENCE IN DIGITAL COMMUNICATION SYSTEMS

Interference is a factor limiting the performance of digital communication systems. Co-channel interference is one of major types of system-generated interference and refers to the degradation caused by an interfering waveform appearing within the signal bandwidth. It can be introduced in a variety of ways, but most commonly by other users of the same portion of RF spectrum operating similar types of equipment [Ref 3]. In Chapter 3 MFSK digital communication systems were modeled, and in Chapter 4 the performance of these MFSK systems operating in the presence AWGN was verified. Since theoretical results for bit error probabilities are available only for the cases of AWGN, one must resort to simulations to estimate the bit error probabilities for MFSK systems with both AWGN and co-channel interference.

In this chapter computer simulation results are provided for BFSK, 4FSK, and 8FSK communication systems operating in the presence of wideband, AWGN and co-channel interference. The results are the averages of two computer simulations for each case. The sampling rate was selected as

twice the minimum sampling rate (an oversampling factor of two), and simulations were run until at least 100 errors were observed. In order to prevent 'out of memory' errors, the data sequences were limited to 10^6 symbols for each simulation. If less than 100 errors were observed in a simulation, the sequence was repeated until a sufficient number of errors occurred. Finally, the computer simulation results were obtained for a range of -5dB to +15 dB symbol signal-to-noise and symbol signal-to-interference (i.e., jamming) ratios.

B. PERFORMANCE OF MFSK DIGITAL COMMUNICATION SYSTEMS IN THE PRESENCE OF AWGN AND CO-CHANNEL INTERFERENCE

In order to specify the performance of MFSK communication systems operating in the presence AWGN and co-channel interference, two sets of plots are provided for M = 2, 4, and 8, for both coherent and noncoherent detection. The bit error probability as a function of the signal-to-noise ratio (SNR) with the signal-to-interference ratio (SJR) as a parameter is shown in the first set, while the second set shows the bit error probability as a function of the signal-to-interference ratio (SJR) with the signal-to-noise ratio (SNR) as a parameter is shown in the second set.

1. Results For Coherent Binary FSK (BFSK)

The probability of bit error for coherent BFSK as a function of the signal-to-noise ratio SNR is shown in Figure 49. The BER increase as SNR decreases, which only happens for negative SJR, is discussed in Appendix E.

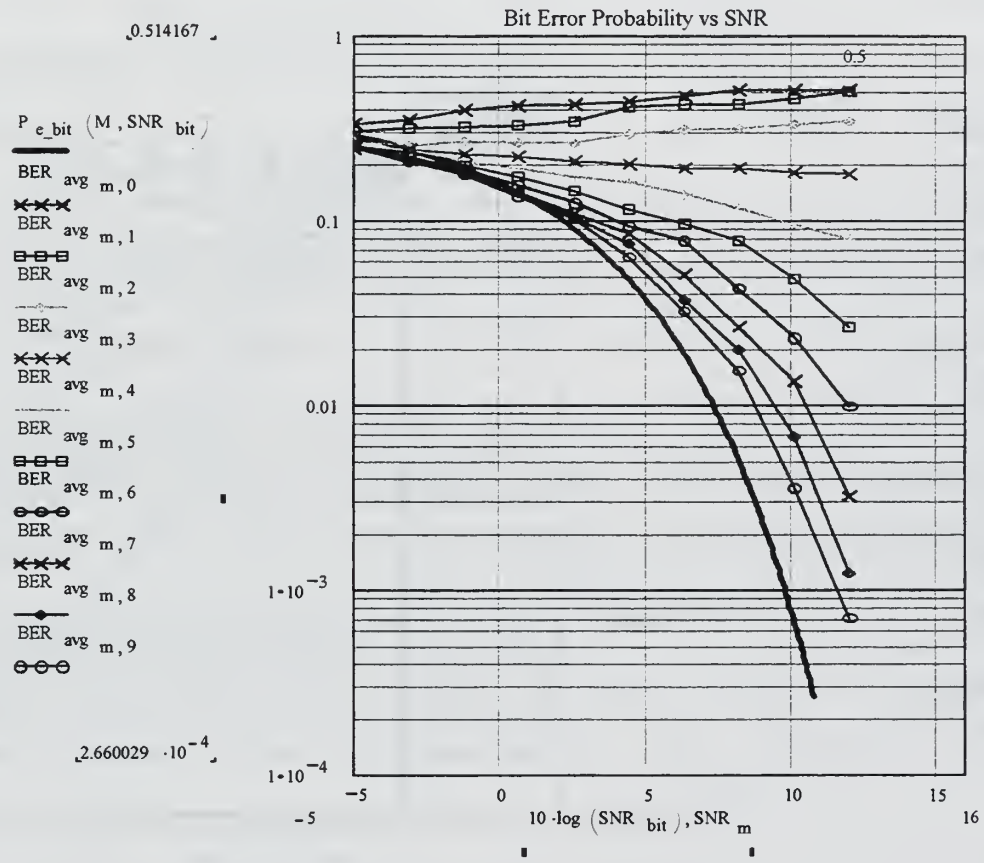

Figure 49. Probability of Bit Error versus SNR for Coherent BFSK with AWGN and Interference.

Ten curves for signal-to-interference (SJR) ratios from -5 dB to +12 dB in 1.89 dB increments are shown in Figure 49. The solid line is the probability of bit error for the BFSK without co-channel interference. We note the dramatic increase in the bit error probability due to co-channel interference relative to the AWGN only.

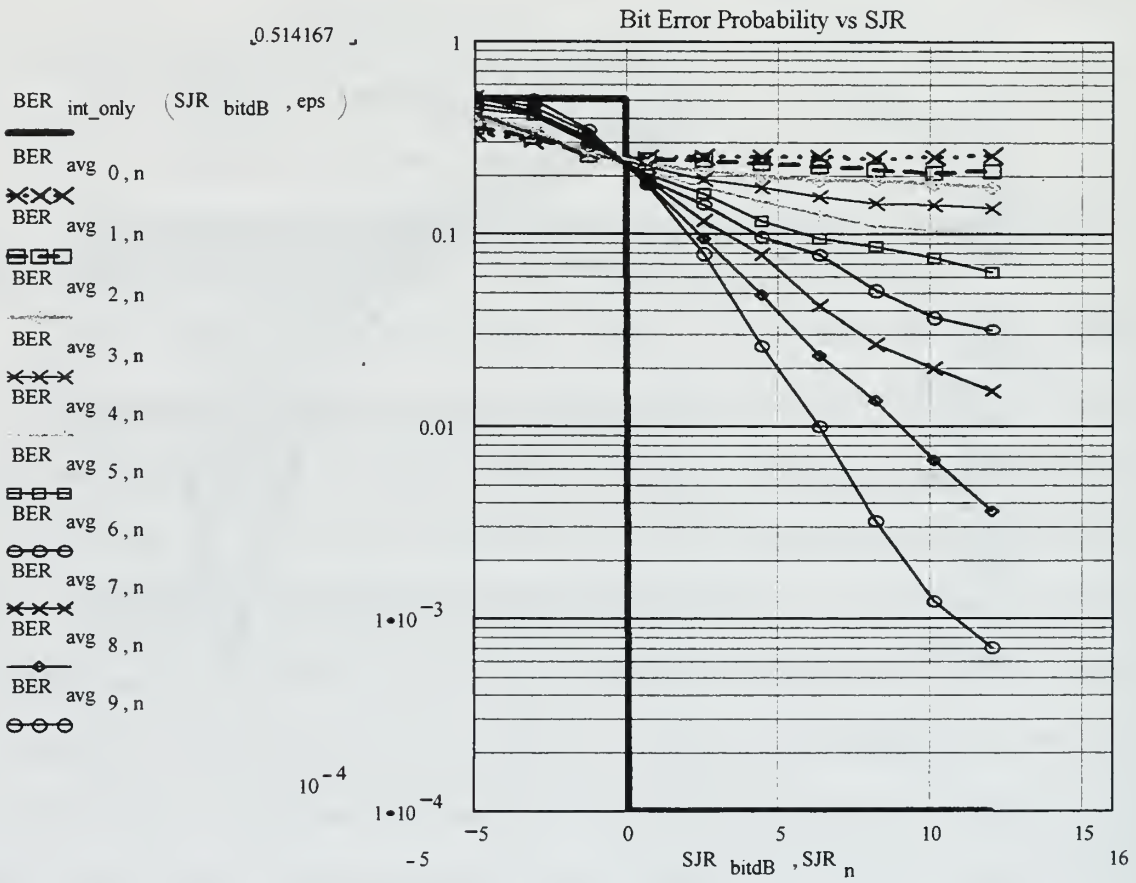

Figure 50. Probability of Bit Error versus SJR for Coherent BFSK with AWGN.

The probability of bit error for coherent BFSK as a function of the signal-to-interference ratio SJR is shown in Figure 50. Ten curves for signal-to-noise (SNR) ratios from -5 dB to +12 dB in 1.89 dB increments are shown. The solid line (unit step function) is the probability of bit error for the case when only signal and interference are present (AWGN). Note that although the unit step has the value of zero for positive SJR, the value of 10^{-4} has been selected for SJR > 0 such that a logarithmic scale that shows the simulation results well (all > 10^{-4}) can be used. When the interference power is less than the signal power, (positive SJR) the interference has no effect, and the probability of error is zero because the receiver always selects the larger of the detected symbols. On the other hand, if the interference power is larger than the signal power, the probability of error is 0.5 since (statistically) half of the interference symbols will be opposite to the signal symbols and will be selected by the receiver because of their larger power. Consequently, the receiver will be making incorrect decisions 50% of the time.

In the noise-free case, the probability of bit errors caused by co-channel interference exhibits step-like behavior (from 0 to 0.5) with the threshold at SJR = 0 dB. As the signal-to-interference ratio increases (that is, as

the interference power decreases relative to the signal power), the probability of bit error decreases at first, but then becomes essentially constant (the curves become nearly horizontal lines). This indicates that there are roughly two regimes of operation: interference-dominated and noise-dominated. Although the curves do not change abruptly from decreasing monotonically to remaining constant, approximate values that separate the two regions may be defined by inspection. For example, the 7-th curve (for SNR=6.333 dB) becomes flat for SJR > 12 dB meaning that the performance of the system is interference dominated for SJR < 12 dB.

In general, the condition SNR = SJR seems appropriate as the border between the interference-dominated (SNR>SJR) and noise-dominated (SJR>SNR), regimes of operation for lower values of the error probability. This is because the curves for the bit error probability are steep for higher values of SNR and SJR, meaning that a small change in SNR or SJR causes a large change in the bit error probability. Therefore, for small values of the bit error probability (larger values of SNR and SJR) the smaller of the two ratios (SNR or SJR) controls the bit error probability.

2. Results For Coherent 4FSK

The probability of bit error for coherent 4FSK as a function of the signal-to-noise ratio SNR is shown in Figure 51.

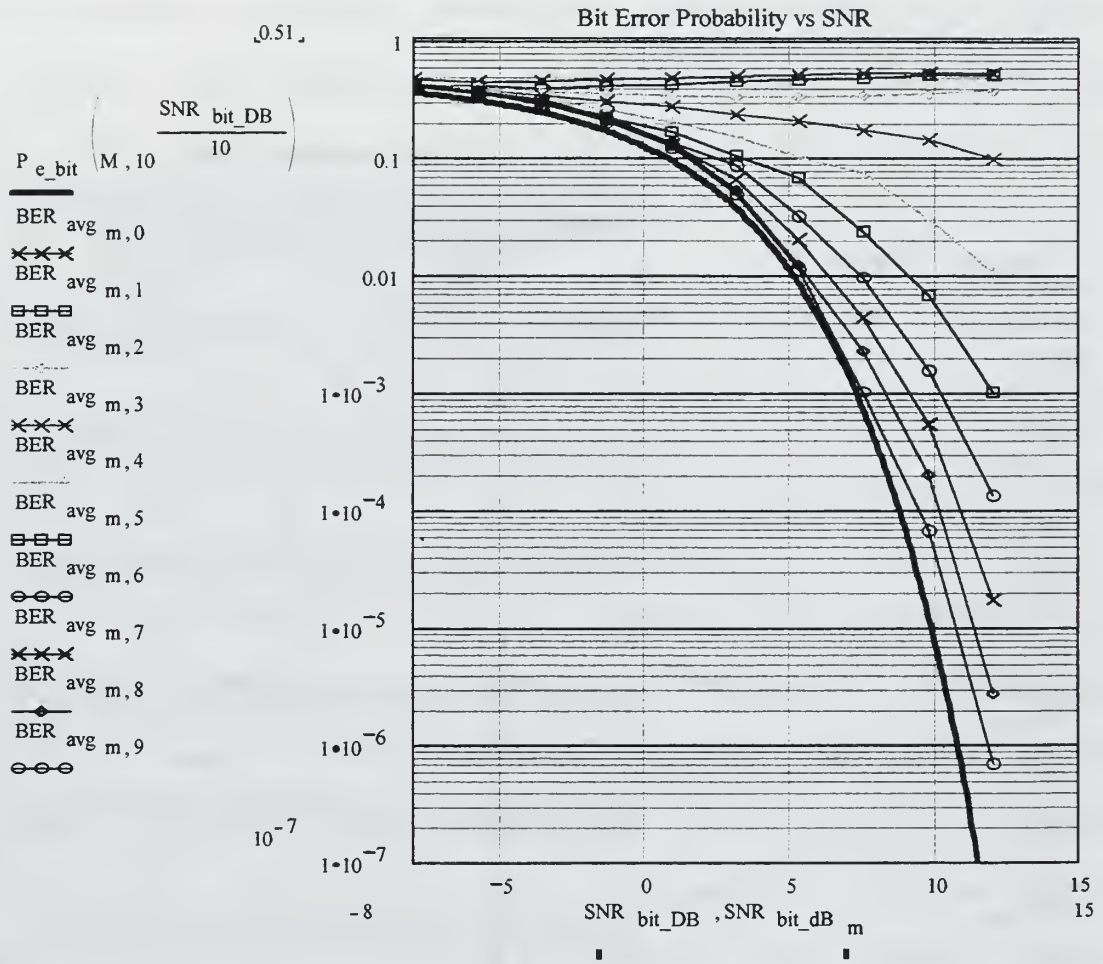

Figure 51. Probability of Bit Error versus SNR for Coherent 4FSK with AWGN and Interference.

Ten curves for symbol signal-to-interference (SJR) ratios from -5 dB to +15 dB in 2.223 dB increments are shown

in Figure 51. The solid line is the probability of error for 4FSK without co-channel interference (AWGN). We note the dramatic increase in the bit error probability due to co-channel interference relative to AWGN only. The probability of bit error for coherent 4FSK as a function of the signal-to-interference ratio SJR is shown in Figure 52.

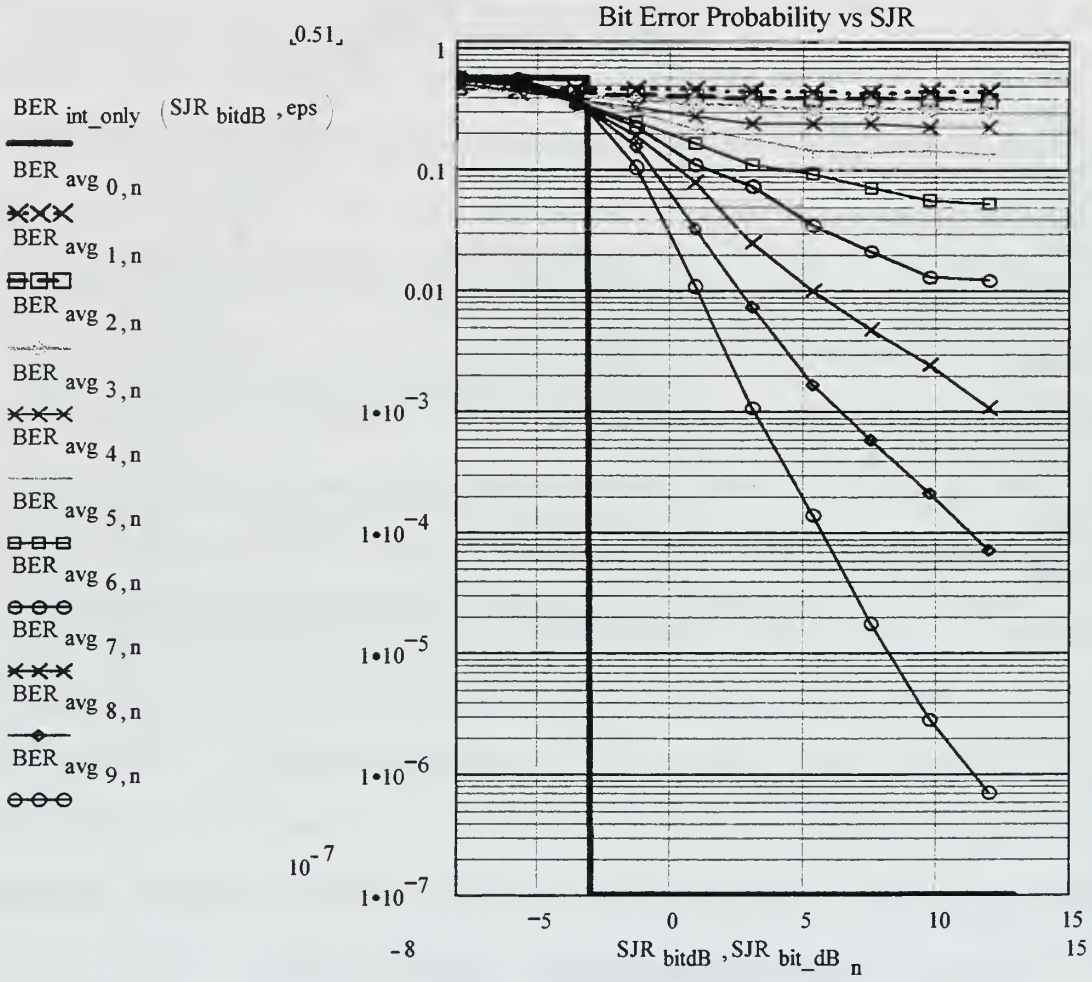

Figure 52. Probability of Bit Error versus SJR for Coherent 4FSK with AWGN and Interference.

Ten curves for symbol signal-to-noise (SNR) ratios from -5 dB to +15 dB in 2.223 dB increments are shown in Figure 52. The solid line (unit step function) shows the probability of bit error for the case when only interference is present (no noise). Since the plots in Figures 50 and 51 represent the probability of bit error for coherent 4FSK, the 'unit step' may also be defined on per-bit basis. For M = 4 the unit step is shifted to the left by a factor of $10 \cdot \log \cdot (\frac{\log(2)}{\log(M)})$ which is approximately 3 dB. Also, although the unit step has the value of zero for positive SJR, the value of 10^{-7} is selected such that a logarithmic scale that shows the simulation results (which are all > 10^{-7}) can be used.

As was found in Section 1 (Results for Coherent BFSK), there are roughly two regimes of operation: interference-dominated and noise-dominated. Although the curves do not change abruptly from decreasing monotonically to being horizontal, approximate values that separate the two regions may be again identified by inspection. For example, the 8-th curve (for the symbol bit SNR=7.545 dB) becomes flat for SJR > 12 dB, meaning that the performance of the system is noise dominated for SJR > 12 dB. In general, as found for coherent BFSK, SNR = SJR is appropriate as the border between the interference-dominated (SNR>SJR) and noise-

dominated (SJR>SNR) regimes of operation for low values of the bit error probability. Therefore, for low values of bit error probability (larger values of SNR and SJR) the smaller of the two (SNR or SJR) by and large controls the bit error probability.

3. Results For Coherent 8FSK

The probability of bit error for coherent 8FSK as a function of the signal-to-noise ratio SNR is shown in Figure 53. Ten curves for symbol signal-to-interference (SJR) ratios from -5 dB to +15 dB are shown. The solid line is the probability of error for the coherent 8FSK without co-channel interference (AWGN). We note the dramatic increase in bit error probability due to co-channel interference relative to AWGN only.

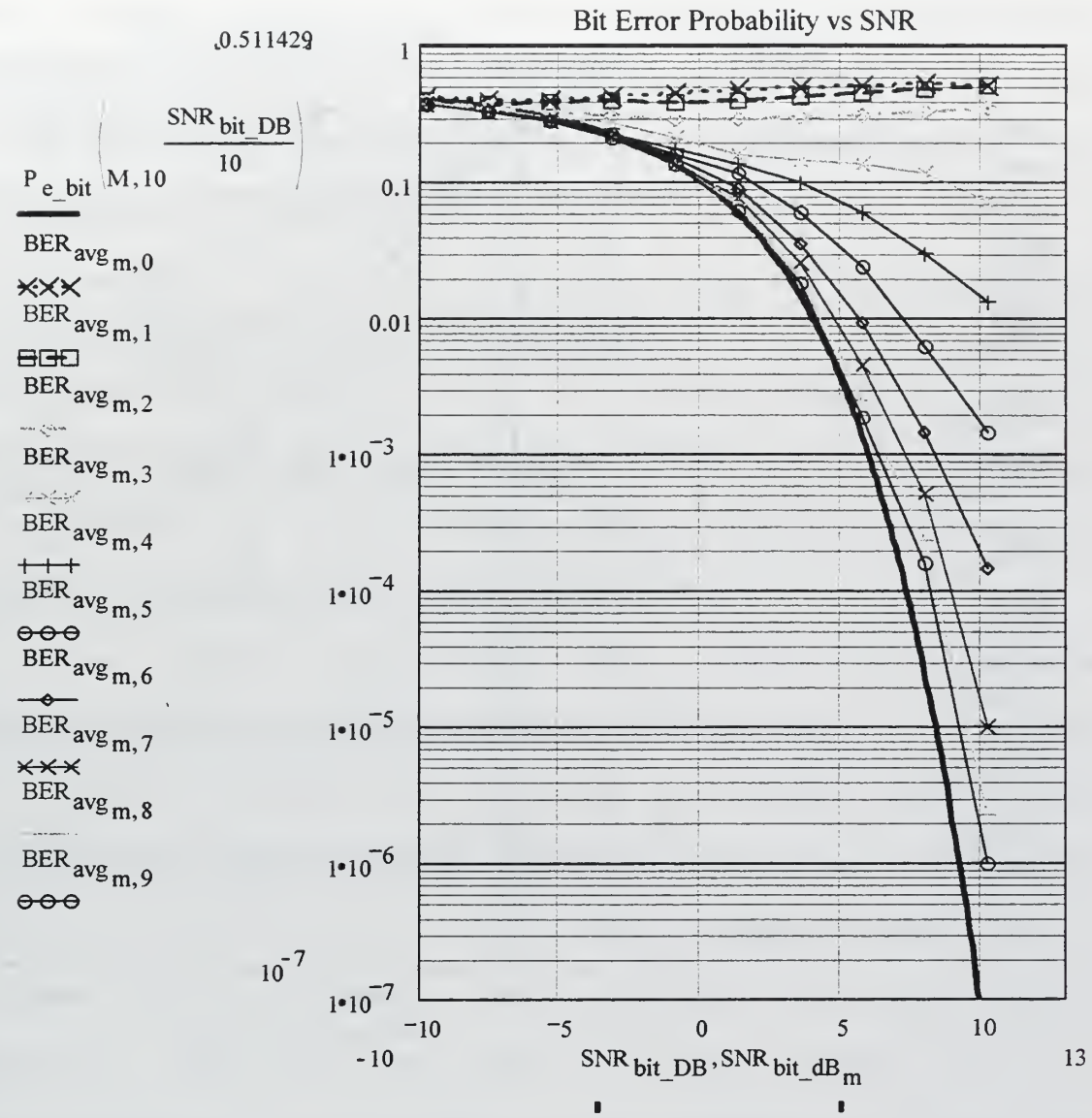

Figure 53. Probability of Bit Error versus SNR for Coherent 8FSK with AWGN and Interference.

The bit error for coherent 8FSK as a function of the signal-to-interference ratio SJR is shown in Figure 54.

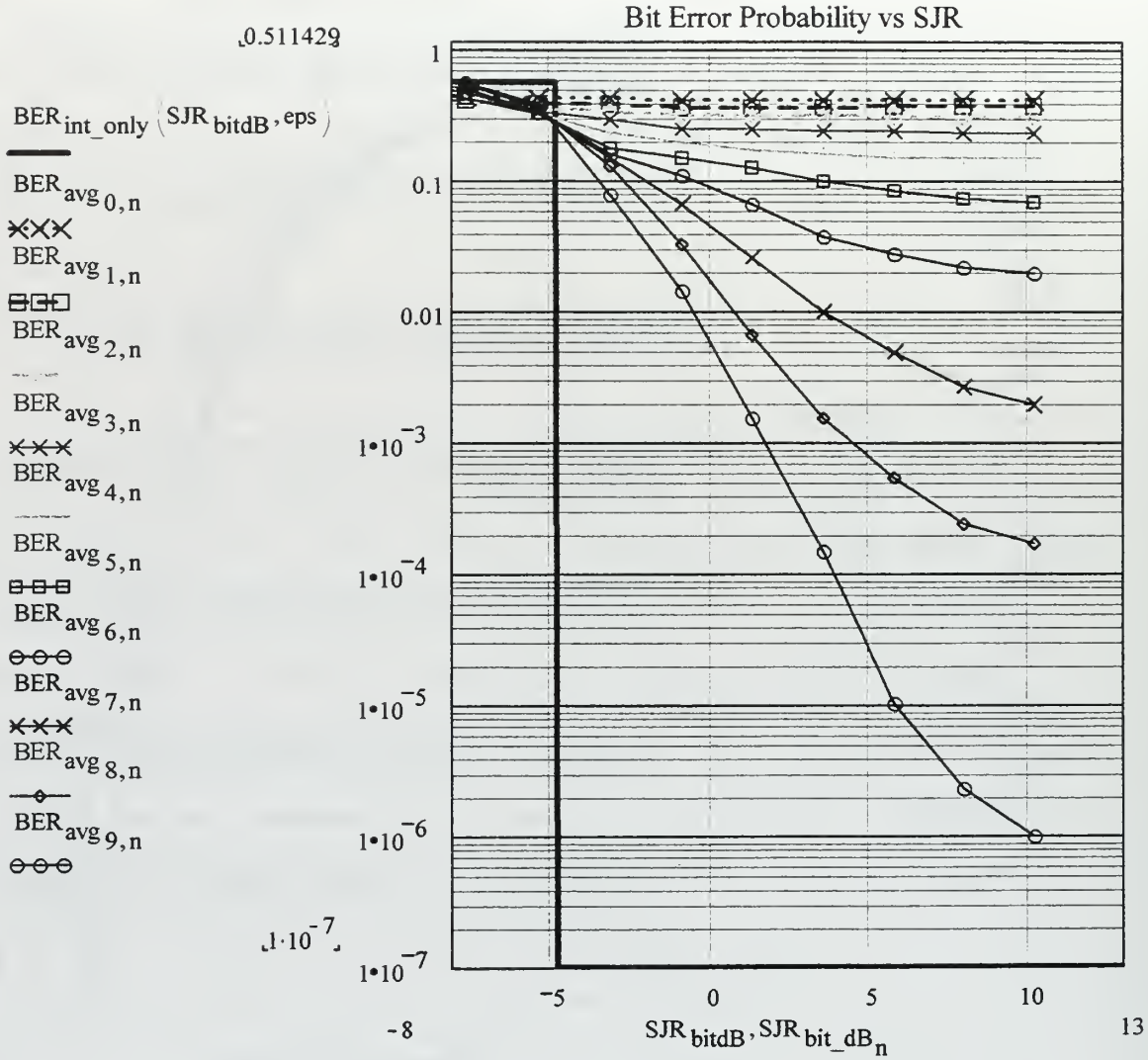

Figure 54. Probability of Bit Error versus SJR for Coherent 8FSK with AWGN and Interference.

Ten curves for symbol signal-to-noise (SNR) ratios from -5 dB to +15 dB in 2.223 dB increments are shown in Figure 54. The solid line (unit step function) is the probability of bit error for the case when only signal and interference

are present (no noise). Since the plots in Figures 53 and 54 represent the probability of bit error for coherent 8FSK, the 'unit step' may also be defined on per-bit basis. For M = 8 the unit step is shifted to the left by a factor of $10 \cdot \log \cdot (\frac{\log(2)}{\log(M)})$, which is approximately 4.71 dB. Also, although the unit step has the value of zero for positive SJR, the value of 10^{-6} has been selected such that a logarithmic scale that best shows the simulation results (which are all $> 10^{-6}$) could be used.

As with the two previous cases, there are roughly two regimes of operation: interference-dominated and noise-dominated. Although the curves do not change abruptly from decreasing monotonically to being horizontal, approximate values that separate the two regions may be identified by inspection. For example, the 8-th curve (for SNR=5.784 dB) becomes flat for bit SJR > 10 dB meaning that the performance of the system is noise dominated for bit SJR > 10 dB. In general, as with the two previous cases, the condition SNR = SJR seems to be the border between the interference-dominated (SNR>SJR) and noise-dominated (SJR>SNR) regimes of operation for lower values of bit error probability. Therefore, for small values of the bit error probability (larger values of SNR and SJR) the smaller of

the two (SNR or SJR) by and large controls the error probability.

4. Results For Noncoherent Binary FSK (BFSK)

The probability of bit error for noncoherent BFSK as a function of the signal-to-noise ratio SNR is shown in Figure 55.

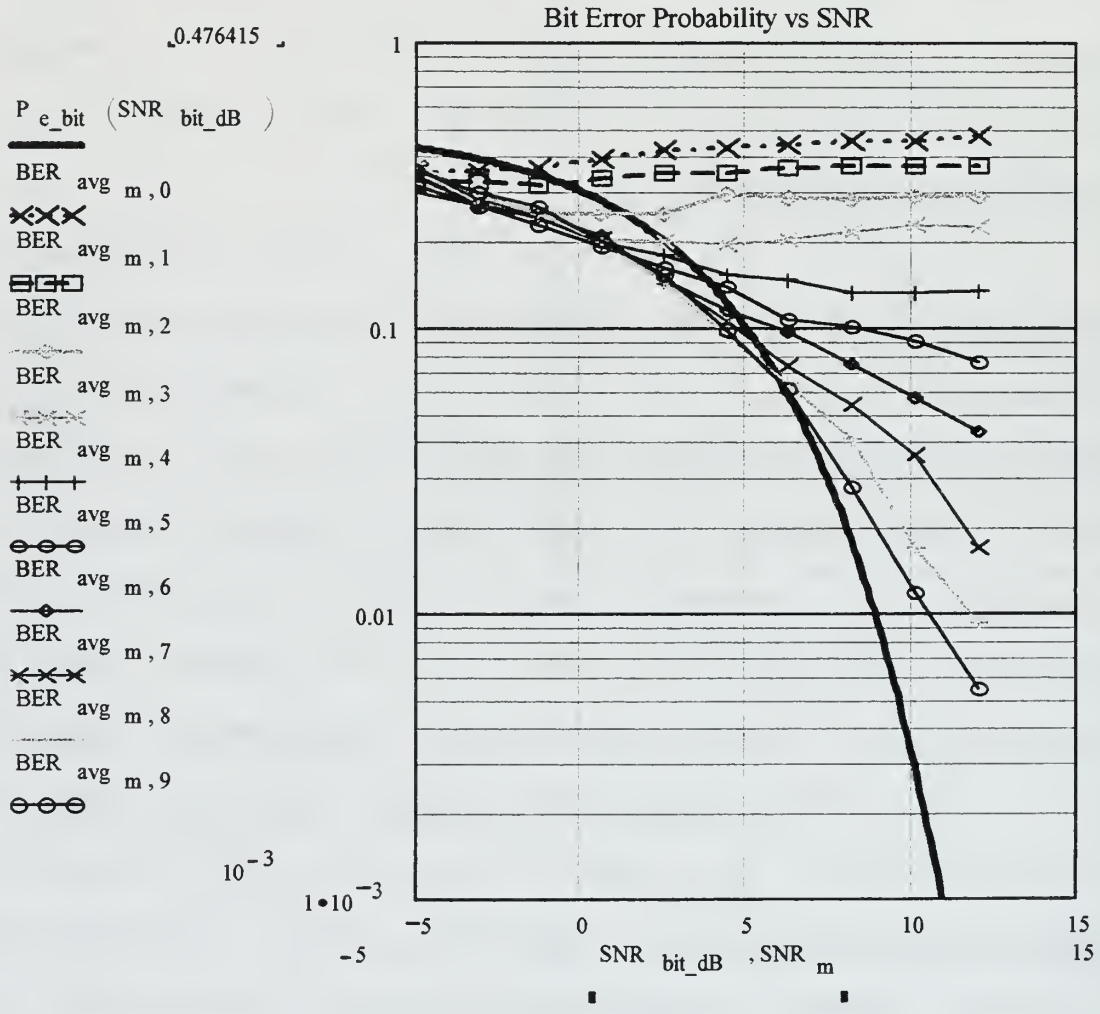

Figure 55. Probability of Bit Error versus SNR for Noncoherent BFSK with AWGN and Interference.

Ten curves for signal-to-interference (SJR) ratios from -5 dB to +12 dB in 1.89 dB increments are shown in Figure 55. The solid line is the probability of bit error for noncoherent BFSK without co-channel interference (AWGN only). We note the dramatic increase in the bit error probability due to co-channel interference relative to AWGN only.

The probability of bit error for noncoherent BFSK as a function of the signal-to-interference ratio SJR is shown in Figure 56. Ten curves for signal-to-noise (SNR) ratios from -5 dB to +12 dB are shown. The solid line (unit step function) is the probability of bit error for the case when only interference is present (no noise). Note that the unit step has the value of zero for positive SJR but the value of 10^{-3} has been selected so that a logarithmic scale that shows the simulation results (which are all > 10^{-3} can be used. As with the previous cases examined, there are roughly two regimes of operation: interference-dominated and noise-dominated. Although the curves do not change abruptly from decreasing monotonically to remaining constant, approximate values that separate the two regions may be identified by inspection. For example, the 7-th curve (for SNR=6.333 dB) becomes flat for SJR > 12 dB meaning that the performance of the system is noise dominated for SJR > 12 dB. In general,

as before, the condition SNR = SJR is the border between the interference-dominated (SNR>SJR) and noise-dominated (SJR>SNR) regimes of operation for lower values of the error probability.

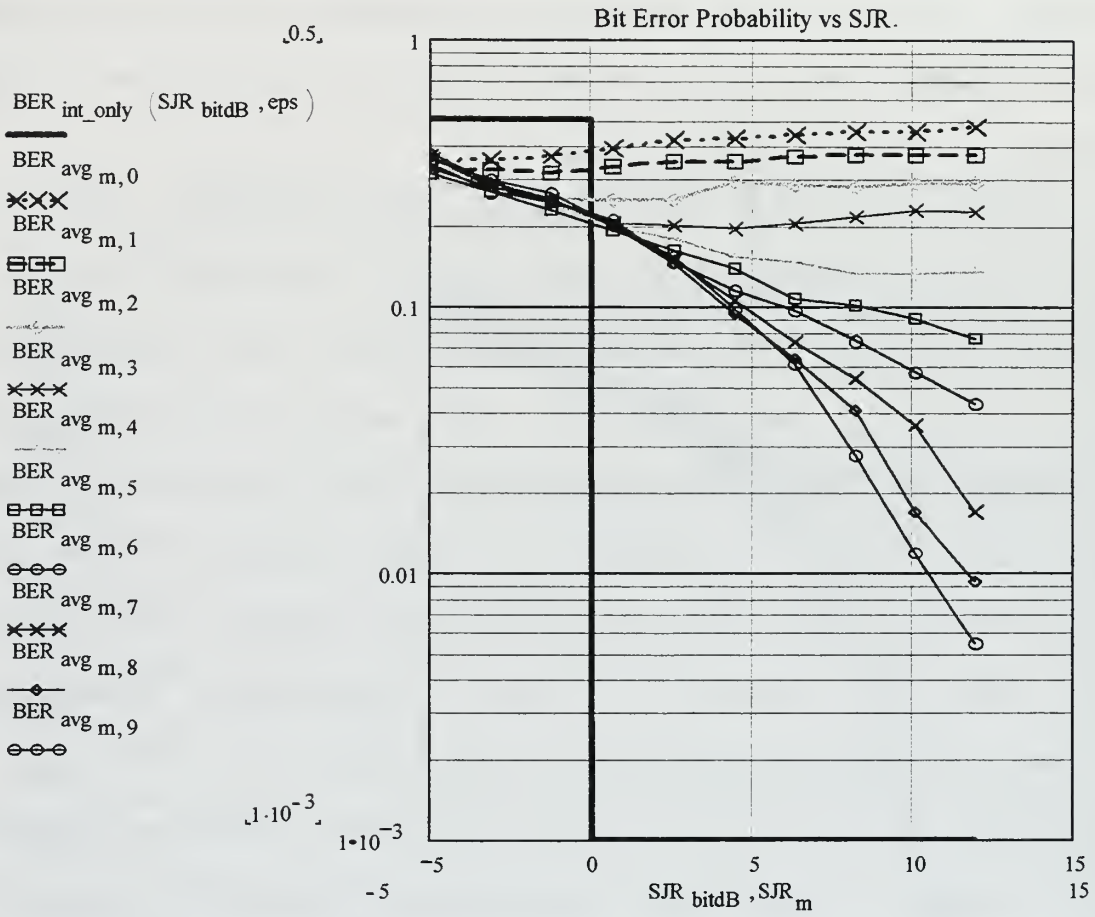

Figure 56. Probability of Bit Error versus SJR for Noncoherent BFSK with AWGN and Interference.

Therefore, for small values of the error probability (larger values of SNR and SJR), the smaller of the two (SNR or SJR) by and large controls the error probability.

5. Results For Noncherent 4FSK

The probability of bit error for noncoherent 4FSK as a function of the signal-to-noise ratio SNR is shown in Figure 57.

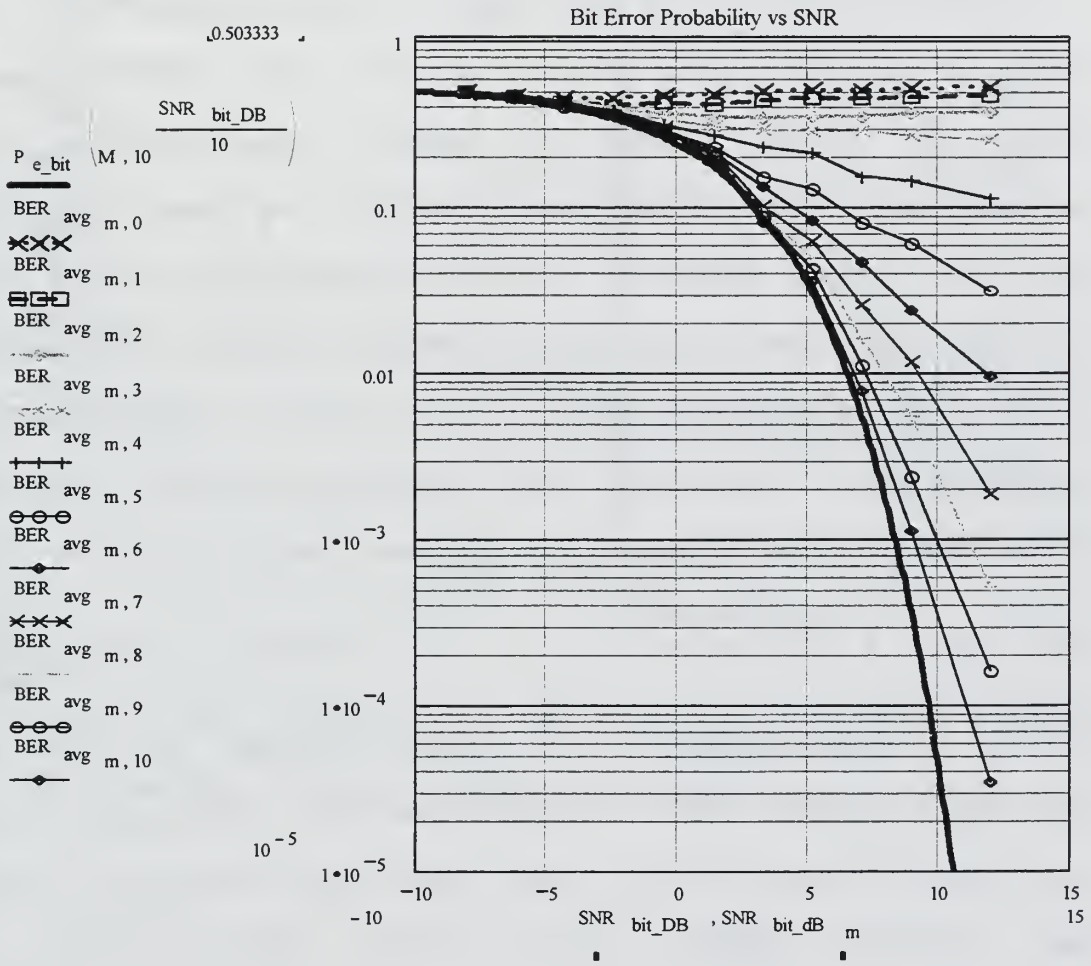

Figure 57. Probability of Bit Error versus SNR for Noncoherent 4FSK with AWGN and Interference.

Eleven curves for symbol signal-to-interference (SJR) ratios from -5 dB to +15 dB in 1.89 dB increments are shown in Figure 57. The solid line is the probability of bit error for the noncoherent 4FSK without co-channel interference (AWGN only). We note the dramatic increase in the bit error probability due to co-channel interference, relative to AWGN only.

The probability of bit error for noncoherent 4FSK as a function of the signal-to-interference ratio SJR is shown in Figure 58. Eleven curves for symbol signal-to-noise (SNR) ratios from -5 dB to +15 dB in 1.89 dB are shown. The solid line (unit step function) is the probability of bit error for the case when only interference is present (no noise). Since the plots in Figures 57 and 58 represent the probability of bit error for noncoherent 4FSK, the 'unit step' may also be defined on per-bit basis. For M = 4 the unit step is shifted to the left by a factor of $10 \cdot \log \cdot (\frac{\log(2)}{\log(M)})$ which is approximately 3 dB. Also, although the unit step has the value of zero for positive SJR, the value of 10^{-5} has been selected so that the logarithmic scale best shows the simulation results (which are all > 10^{-5}). As with the previous cases examined, there are roughly two regimes of operation: interference-dominated and noise-dominated.

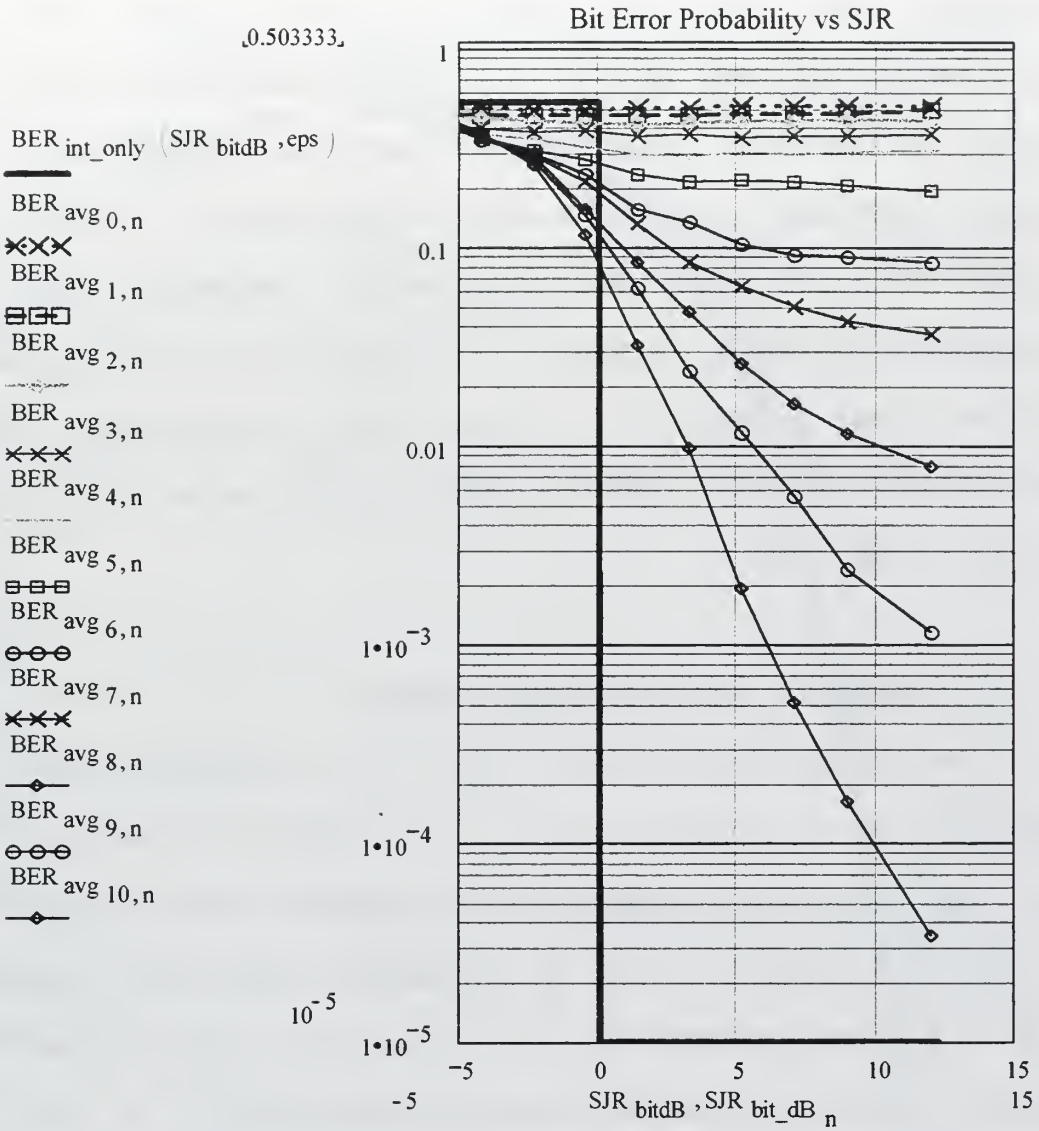

Figure 58. Probability of Bit Error versus SJR for Noncoherent 4FSK with AWGN and Interference.

Although the curves do not change abruptly from decreasing monotonically to remaining constant, approximate values that separate the two regions may be identified by inspection. For example, the 9-th curve (for SNR=8.99 dB)

becomes flat for bit SJR > 12 dB, meaning that the performance of the system is noise dominated for bit SJR > 12 dB. In general, as with previous cases, SNR = SJR is the border between the interference-dominated (SNR>SJR) and noise-dominated (SJR>SNR) regimes of operation for lower values of the error probability. Therefore, for small values of the error probability (larger values of SNR and SJR) the smaller of the two (SNR or SJR) by and large controls the error probability.

6. Results For Noncoherent 8FSK

The probability of bit error for noncoherent 8FSK as a function of the signal-to-noise ratio SNR is shown in Figure 59. Ten curves for signal-to-interference (SJR) ratios from -5 dB to +15 dB in 2.223 dB increments are shown. The solid line is the probability of error for noncoherent 8FSK without co-chanel interference (AWGN only). We note the dramatic increase in the bit error probability due to co-channel interference relative to the AWGN only.

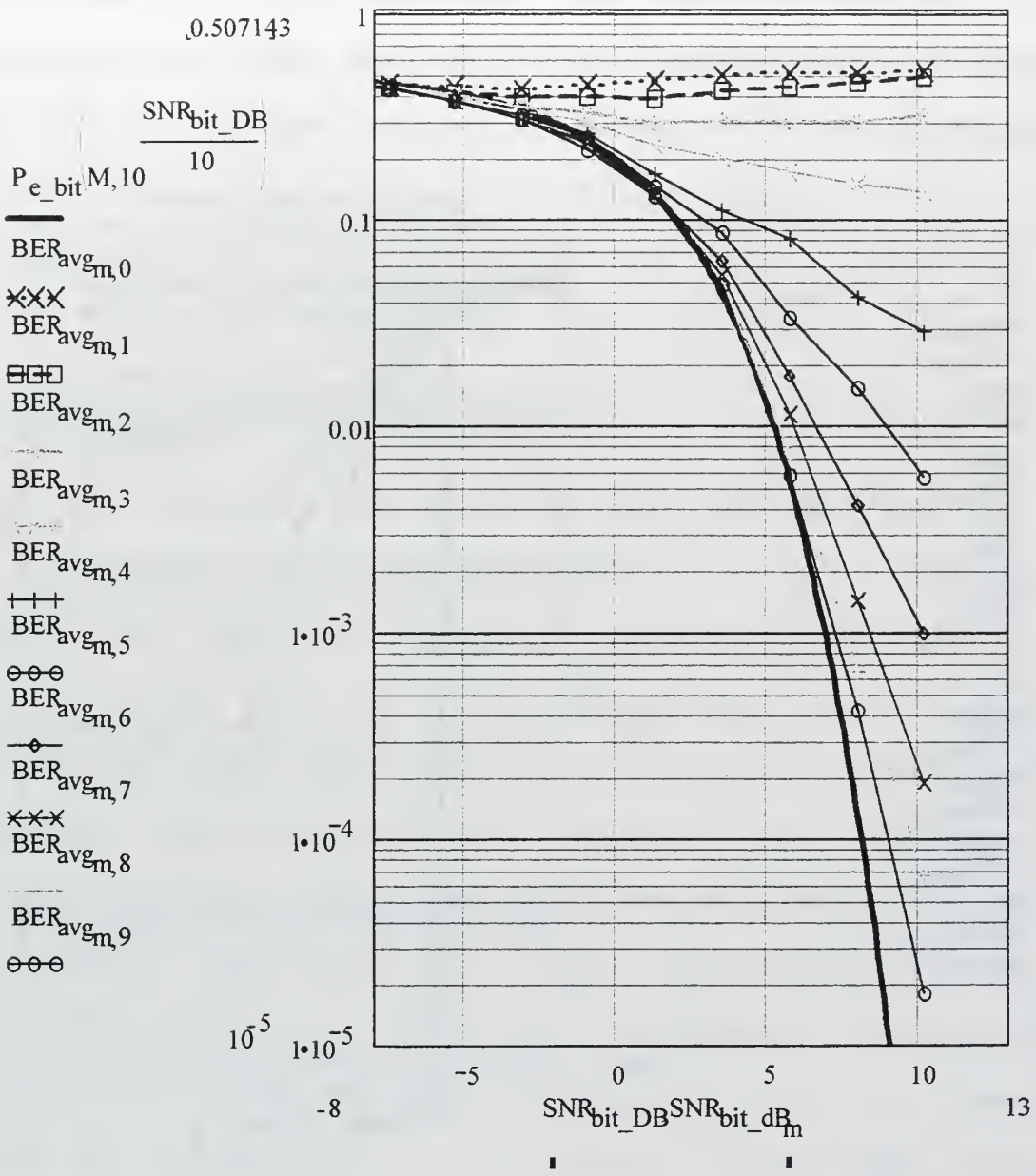

Figure 59. Probability of Bit Error versus SNR for Noncoherent 8FSK with AWGN and Interference.

The probability of bit error for noncoherent 8FSK as a function of the signal-to-interference ratio SJR is shown in Figure 60.

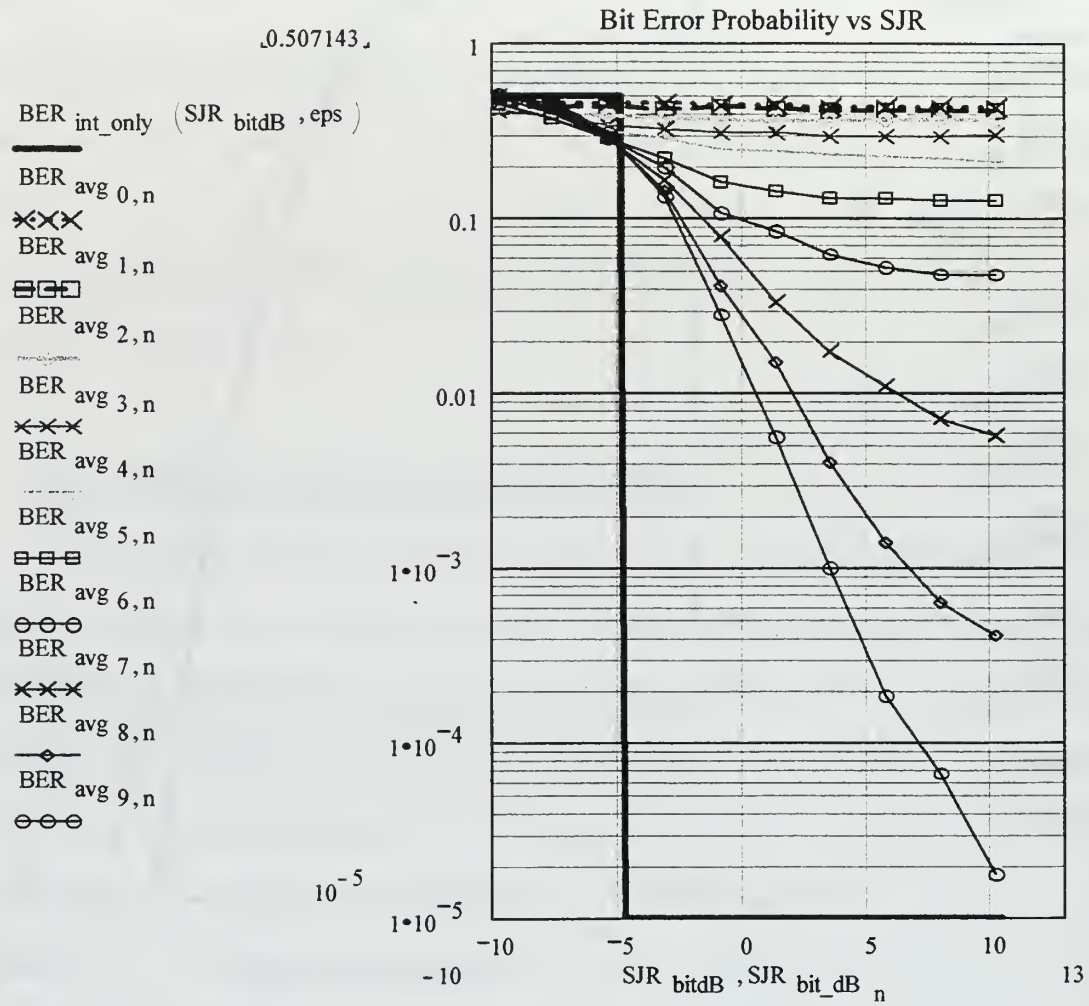

Figure 60. Probability of Bit Error versus SJR for Noncoherent 8FSK with AWGN and Interference.

Ten curves for symbol signal-to-noise (SNR) ratios from -5 dB to +15 dB in 2.223 dB increments are shown in Figure

60. The solid line (step function) shows the probability of bit error for the cases when only interference is present (no noise). Since the plots in Figures 58 and 59 represent the probability of bit error for noncoherent 8FSK the 'unit step' may also be defined on per-bit basis. For M = 8 the unit step is shifted to the left by a factor of $10 \cdot \log \cdot (\frac{\log(2)}{\log(M)})$, which is approximately 4.71 dB. Also, although the unit step has the value of zero for positive SJR the value of 10^{-5} has been selected such that a logarithmic scale that shows the simulation results (which are all > 10^{-5}) can be used. As with the previous cases examined, there are roughly two regimes of operation: interference-dominated and noise-dominated. Although the curves do not change abruptly from decreasing monotonically to being horizontal, approximate values that separate the two regions may be identified by inspection. For example, the 9-th curve (for SNR=8.007 dB) becomes flat for bit SJR > 10 dB meaning that the performance of the system is noise dominated for bit SJR > 10 dB. In general, as before, SNR=SJR is the border between the interference-dominated (SNR>SJR) and noise-dominated (SJR>SNR) regimes of operation for lower values of the error probability. Therefore, for small values of the error probability (larger values of SNR and SJR) the

smaller of the two (SNR or SJR) by and large controls the bit error probability.

C. OBSERVATIONS

The simulation results for the bit error probability as a function of SNR with SJR as a parameter can be used to determine the increase in bit error probability due to co-channel interference (referred here as 'jamming'). In the case of intentional jamming, the objective is to increase the number of bit errors. In such a case, the jammer needs to minimize the receiver's signal-to-interference ratio as much as possible. The maximum signal-to-interference ratio that still allows for the desired increase in the probability of bit errors can be determined from the simulation results. Given the desired jamming-generated increase in bit error probability, the required jammer power can be calculated (also given the other relevant parameters such as the distance from the target receiver, the receiver antenna gain in the direction of the jammer, and the like are also known). For example, from the curves for the noncoherent 8FSK case, a -10 dB signal-to-interference ratio results in a bit error probability of 0.5, which means that the communications is essentially impossible. As SJR increases, the bit error probability decreases. Nevertheless, bit error probability for both noise and

interference is always larger than the bit error probability for noise alone. For example, for SJR=10.23 dB and SNR=10.23 dB, the bit error probability is about 20 times larger than the bit error probability for noise alone.

In summary, assuming that the operational SNR for the MFSK communication channel is known, the family of curves of P_e versus SNR with SJR as a parameter can be used to determine the increase in the probability of error due to various levels of co-channel interference. In the case of intentional jamming with another synchronized MFSK signal, the curves may be used to determine the jammer-on-target requirements for a certain increase in bit error probability. Subsequently, for a given scenario (jammer-to-target distance, etc.) the jammer requirements such as jammer power, antenna gain, etc. can be calculated.

The second set of curves gives the bit error probability of MFSK communication systems in the presence of both noise and co-channel interference as a function of SJR and with SNR ratio as a parameter. The effect of co-channel interference is to introduce errors in transmission that are quantified by the symbol error probability P_e or the Bit Error Rate P_b (BER). For BFSK the largest bit error probability is 0.5 or 50%. For MFSK the largest symbol error probability is given by:

$$\text{Max } (P_e) = 1 - 1/M \qquad (5.1)$$

where M is the number of symbols. The maximum bit error probability is given by:

$$\text{Max } (P_b) = \frac{1}{2}\frac{M}{M-1}\left(1-\frac{1}{M}\right)=0.5 \qquad (5.2)$$

Therefore, the maximum bit error probability for MFSK is also 0.5, and in the noise-free case (SNR tends to infinity), the theoretical bit error probability due to co-channel interference is a unit step function. The curves for bit error probabilities when noise and co-channel interference are both present tend to 0.5 for negative signal-to-interference ratios (in dB) and tend asymptotically to zero (no errors) for large positive SJR ratios. The curves for negative SNR's are essentially 'flat' since the noise 'controls' the errors. On the other hand, if the SNR is large and positive (the noise power is small), the curves tend to the 'unit step' function of the interference-only case. The most realistic cases of comparable SNR and SJR values are those that fall in between the 'flat' curves at high (close to 0.5) values of

probability of error and the 'unit step' function of the interference-only case.

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

In this thesis MFSK communication systems with coherent and noncoherent detection were simulated in the time domain using MATLAB Simulink and the Communications Toolbox. The bit error probabilities for coherent and noncoherent detection of BFSK, 4FSK, and 8FSK have been determined by simulation and verified against the theoretical values for the case of AWGN. The MFSK models were modified to include the co-channel interference and the bit error probabilities obtained for the ranges SNR and SJR from -5 dB to +15 dB.

Simulation results for coherent detection of BFSK, 4FSK, and 8FSK in the presence of AWGN show excellent agreement with theoretical results. On the other hand, simulation results for the probability of bit error for noncoherent detection of BFSK, 4FSK, and 8FSK differ from the theoretical probabilities of bit error by approximately -30% for noncoherent BFSK, -20% for noncoherent 4FSK and -13% for noncoherent 8FSK, indicating a 'systematic' error in the Communications Toolbox implementation of the noncoherent MFSK detection. Finally, for symbol signal-to-noise and signal-to-interference ratios larger than 15 dB (SNRs and

SJR > 15 dB), the required computational time can be excessive for 100 or so errors (days or even weeks on a 200 MHz PC).

B. RECOMMENDATIONS

The continuation of the research may include:

- Derivation of the theoretical expression for the bit error probability for MFSK with AWGN and co-channel interference (BER as a function of SNR and SJR) for the following cases:
  - phase-locked co-channel interference (relative difference between the interference and signal equal 0),
  - random phase of the co-channel interference (relative to the signal) with uniform distribution between 0 and 2π, and
  - random phase of the co-channel interference with Gaussian distribution.
- Verification of the theoretical bit error probabilities for the above cases using Matlab/Simulink models for Monte Carlo type simulations.

APPENDIX A. MATLAB PROGRAM SUN_PREP1.M FOR COHERENT MFSK CASE.

This program prepares the data file for simulation runs in case of coherent detection of a MFSK communication system in presence of both additive Gaussian noise and co-channel interference.

```
clear

noise_only = menu(' Select: ',...
                  ' NOISE',...
                  ' NOISE and INTERFERENCE ');

num_levels  = input('Enter the number of frequencies M [2]:
');
if isempty(num_levels), num_levels = 2; end

T_sym       = input('Enter the symbol duration T [1]: ');
if isempty(T_sym), T_sym = 1; end

oversampling = input('Enter the oversampling factor [2]: ');
if isempty(oversampling), oversampling = 2; end

min_SNR = input('Enter the MIN Signal to Noise ratio[-5 dB]:
');
if isempty(min_SNR), min_SNR = -5; end

max_SNR = input('Enter the MAX Signal to Noise ratio [15
dB]: ');
if isempty(max_SNR), max_SNR = 15; end

if noise_only ~= 1
   min_SJR = input('Enter the MIN Signal to Interference
ratio [-5 dB]: ');
   if isempty(min_SJR), min_SJR = -5; end
   max_SJR = input('Enter the MAX Signal to Interference
ratio[10 dB]: ');
    if isempty(max_SJR), max_SJR = 10; end
end

if min_SNR == max_SNR
```

```
   num_noise = 1;
else
  num_noise = input('Enter the number of values for SNR
[10]: ');
   if isempty(num_noise), num_noise = 10; end
end

if noise_only ~= 1
   if min_SJR == max_SJR
      num_jam = 1;
   else
     num_jam = input('Enter the number of values for SJR
[10]: ');
      if isempty(num_jam), num_jam = 10; end
   end
else
   num_jam = 1;
end

min_errors = input('Enter the min number of errors
acceptable [100]: ');
if isempty(min_errors), min_errors = 100; end

error_factor = input('Enter the factor multiplying the
number of errors [2]: ');
if isempty(error_factor), error_factor = 2; end

initial_num_symbols = error_factor*min_errors;

max_randint = input('Enter the maximum size of the random
integer arrays [10^6]: ');
if isempty(max_randint), max_randint = 10^6; end

file_name = input('Enter the file name to save data [no
ext]: ','s');

save sun_data1
```

APPENDIX B. MATLAB PROGRAM SUN_FSK1.M FOR COHERENT MFSK CASE

This program runs coherent MFSK co-channel interference with additive Gaussian noise.

```
clear
load sun_dat1  %This loads all the input data
delta_freq  = 1/T_sym
f_max       = (num_levels-1)*1/T_sym;
delta_t     = 0.5/(f_max*oversampling)
seeds       = randint(3,1,1000);
signal_seed = seeds(1);
noise_seed  = seeds(2);
interf_seed = seeds(3);
initial_num_symbols = error_factor*min_errors;
tic

if num_noise > 1
   delta_SNR = (max_SNR - min_SNR) / (num_noise -1);
else
   delta_SNR = 0;
end

SNR = min_SNR + [0:num_noise - 1]*delta_SNR;
noise_var_vect   = T_sym/(2*delta_t) .* 10 .^ (-SNR/10);

if noise_only ~= 1
   if num_jam > 1
      delta_SJR = (max_SJR - min_SJR) / (num_jam -1);
   else
      delta_SJR = 0;
   end
   SJR = min_SJR + [0:num_jam   - 1]*delta_SJR;
   interf_gain_vect = 10 .^ (-SJR/20);
else
   SJR = - 100;            % There is no Jamming so the SJR
in dB is -infinity
   interf_gain_vect = 0;
end

BER = zeros(num_noise,num_jam);

total_symbols = initial_num_symbols;

for noise_case = 1:num_noise

   noise_var = noise_var_vect(noise_case);
```

```
   for jam_case = 1:num_jam

        enough_errors = 0;
        num_err       = 0;
        num_symbols   = initial_num_symbols;
        rand_int      = min([num_symbols max_randint]);
        total_symbols = num_symbols

        while enough_errors ~= 1
            interf_gain = interf_gain_vect(jam_case);
            clear error_number
                    sim('fskco_bm',num_symbols)
               [new_errors err_cols] = size(error_number);
               num_err = num_err + new_errors;
               if num_err == 0
                 num_symbols   = num_symbols*min_errors;
                 rand_int      = min([num_symbols
max_randint]);
                 total_symbols = total_symbols +
num_symbols;
               elseif (num_err > 0 & num_err < min_errors)
                 num_symbols   = (min_errors -
num_err)*ceil(total_symbols/num_err);
                 total_symbols = total_symbols +
num_symbols;
            else
                 enough_errors = 1;
               end
        end

        number_of_errors(noise_case,jam_case)  = num_err
        number_of_symbols(noise_case,jam_case) =
total_symbols
        BER(noise_case,jam_case)               = num_err /
total_symbols
     end

     eval([' save ' file_name '.snr SNR -ascii']);
     eval([' save ' file_name '.sjr SJR -ascii']);
     eval([' save ' file_name '.ber BER -ascii']);
     eval([' save ' file_name '.ner number_of_errors -
ascii']);
     eval([' save ' file_name '.nsy number_of_symbols -
ascii']);

end
toc
```

APPENDIX C. MATLAB PROGRAM SUN_PNC1.M FOR NON-COHERENT MFSK CASE

This program prepares the data file for simulation runs in case of non-coherent detection of a MFSK communication system in presence of both additive Gaussian noise and co-channel interference.

```
%%% This prepares the data file for sun runs

clear

noise_only = menu(' Select: ',...
                  ' NOISE',...
                  ' NOISE and INTERFERENCE ');

num_levels  = input('Enter the number of frequencies M [2]:
');
if isempty(num_levels), num_levels = 2; end

T_sym       = input('Enter the symbol duration T [1]: ');
if isempty(T_sym), T_sym = 1; end

oversampling = input('Enter the oversampling factor [2]: ');
if isempty(oversampling), oversampling = 2; end

min_SNR = input('Enter the MIN Signal to Noise ratio[-5 dB]:
');
if isempty(min_SNR), min_SNR = -5; end

max_SNR = input('Enter the MAX Signal to Noise ratio [15
dB]: ');
if isempty(max_SNR), max_SNR = 15; end

if noise_only ~= 1
  min_SJR = input('Enter the MIN Signal to Interference
ratio [-5 dB]: ');
  if isempty(min_SJR), min_SJR = -5; end
  max_SJR = input('Enter the MAX Signal to Interference
ratio[10 dB]: ');
   if isempty(max_SJR), max_SJR = 10; end
end

if min_SNR == max_SNR
```

```
   num_noise = 1;
else
  num_noise = input('Enter the number of values for SNR
[10]: ');
   if isempty(num_noise), num_noise = 10; end
end

if noise_only ~= 1
   if min_SJR == max_SJR
      num_jam = 1;
   else
     num_jam = input('Enter the number of values for SJR
[10]: ');
      if isempty(num_jam), num_jam = 10; end
   end
else
   num_jam = 1;
end

min_errors = input('Enter the min number of errors
acceptable [100]: ');
if isempty(min_errors), min_errors = 100; end

error_factor = input('Enter the factor multiplying the
number of errors [2]: ');
if isempty(error_factor), error_factor = 2; end

initial_num_symbols = error_factor*min_errors;

max_randint = input('Enter the maximum size of the random
integer arrays [10^6]: ');
if isempty(max_randint), max_randint = 10^6; end

file_name = input('Enter the file name to save data [no
ext]: ','s');

save sun_dnc1
```

APPENDIX D. MATLAB PROGRAM SUN_NC1.M FOR NON-COHERENT MFSK CASE

This program runs coherent MFSK co-channel interference with additive Gaussian noise.

```
clear

load sun_dnc1  %This loads all the input data

delta_freq  = 1/T_sym
f_max       = (num_levels-1)*1/T_sym;
delta_t     = 0.5/(f_max*oversampling)
seeds       = randint(3,1,1000);
signal_seed = seeds(1);
noise_seed  = seeds(2);
interf_seed = seeds(3);
initial_num_symbols = error_factor*min_errors;

tic

if num_noise > 1
   delta_SNR = (max_SNR - min_SNR) / (num_noise -1);
else
   delta_SNR = 0;
```

```
end

SNR = min_SNR + [0:num_noise - 1]*delta_SNR;
noise_var_vect    = T_sym/(2*delta_t) * 10 .^ (-SNR/10);

if noise_only ~= 1
   if num_jam > 1
      delta_SJR = (max_SJR - min_SJR) / (num_jam -1);
   else
      delta_SJR = 0;
   end
   SJR = min_SJR + [0:num_jam    - 1]*delta_SJR;
   interf_gain_vect = 10 .^ (-SJR/20);
else
   SJR = - 100;              % There is no Jamming so the SJR
in dB is -infinity
   interf_gain_vect = 0;
end

BER = zeros(num_noise,num_jam);

total_symbols = initial_num_symbols;

for noise_case = 1:num_noise
```

```
    noise_var = noise_var_vect(noise_case);

    for jam_case = 1:num_jam

        enough_errors = 0;
        num_err       = 0;
        num_symbols   = initial_num_symbols;
        rand_int      = min([num_symbols max_randint]);
        total_symbols = num_symbols

        while enough_errors ~= 1
            interf_gain = interf_gain_vect(jam_case);
            clear error_number
            sim('fsknc_bm',num_symbols)
            [new_errors err_cols] = size(error_number);
            num_err = num_err + new_errors;
            if num_err == 0
                num_symbols   = num_symbols*min_errors;
                rand_int      = min([num_symbols
max_randint]);
                total_symbols = total_symbols +
num_symbols;
            elseif (num_err > 0 & num_err < min_errors)
                num_symbols   = (min_errors -
num_err)*ceil(total_symbols/num_err);
```

```
                total_symbols = total_symbols +
num_symbols;

            else

                enough_errors = 1;

            end

        end


        number_of_errors(noise_case,jam_case)  = num_err

        number_of_symbols(noise_case,jam_case) =
total_symbols

        BER(noise_case,jam_case)               = num_err /
total_symbols


    end

    eval([' save ' file_name '.snr SNR -ascii']);

    eval([' save ' file_name '.sjr SJR -ascii']);

    eval([' save ' file_name '.ber BER -ascii']);

    eval([' save ' file_name '.ner number_of_errors -
ascii']);

    eval([' save ' file_name '.nsy number_of_symbols -
ascii']);

end


toc
```

APPENDIX E. THEORETICAL BER FOR COHERENT BFSK WITH AWGN AND CO-CHANNEL INTERFERENCE

The BER as a function of SNR for negative values of SJR (interference power exceeds the signal power) exhibit somewhat counter-intuitive behaviour. Namely, the BER *increases* as the SNR increases (there are more errors for lower noise power levels). In order to resolve whether or not this is an artifact of the simulation, we have derived the expression for the theoretical BER for coherent BFSK with AWGN and co-channel interference which is assumed to be coherent with the signal. This result of the derivation is:

$$P_{Br} = \frac{1}{2}\left[1 + Q\left(\frac{10^{\frac{-SJR}{20}} + 1}{\frac{10^{\frac{-SNR}{20}}}{\sqrt{2}}}\right) - Q\left(\frac{10^{\frac{-SJR}{20}} - 1}{\frac{10^{\frac{-SNR}{20}}}{\sqrt{2}}}\right)\right] \qquad \text{(E.1)}$$

where SJR and SNR are assumed to be in dB.

Equation (E.1) allows us to plot BER as a surface with SNR and SJR as variables, as shown in Figure 60, for the ranges of SNR and SJR between -20dB and +20dB.

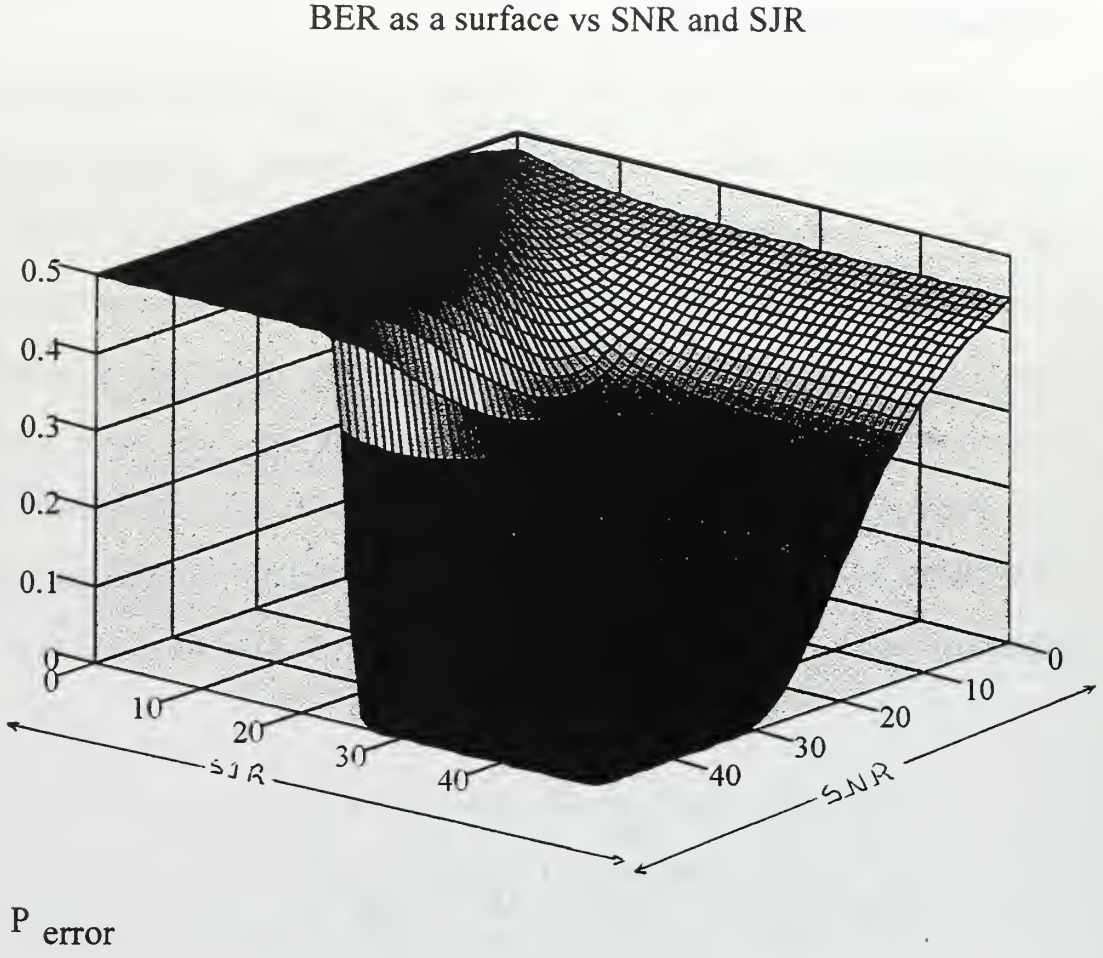

Figure 60. Probability of Bit Error versus SNR and SJR for Coherent BFSK with AWGN and Interference.

The probability of bit error as a function of SNR and SJR shows a region of shallow local minima with the probability of bit error decreasing as the noise power increases. This occurs only for high values of interference and noise and only up to an SJR of 0 dB. For higher values

of SJR there is no local minimum, the decrease with SNR becomes monotonic. The conclusion is that noise actually reduces the probability of bit error at low SJR. This is mainly of academic interest since the reduction occurs at very high values of error probability (from about 0.5 to about 0.35).

Equation (E.1) can also be used to generate conventional BER plots. These plots represent cross-sections of the surface illustrated in Figure 60 along planes of constant SNR, shown in Figure 61, or constant SJR, shown in Figure 62.

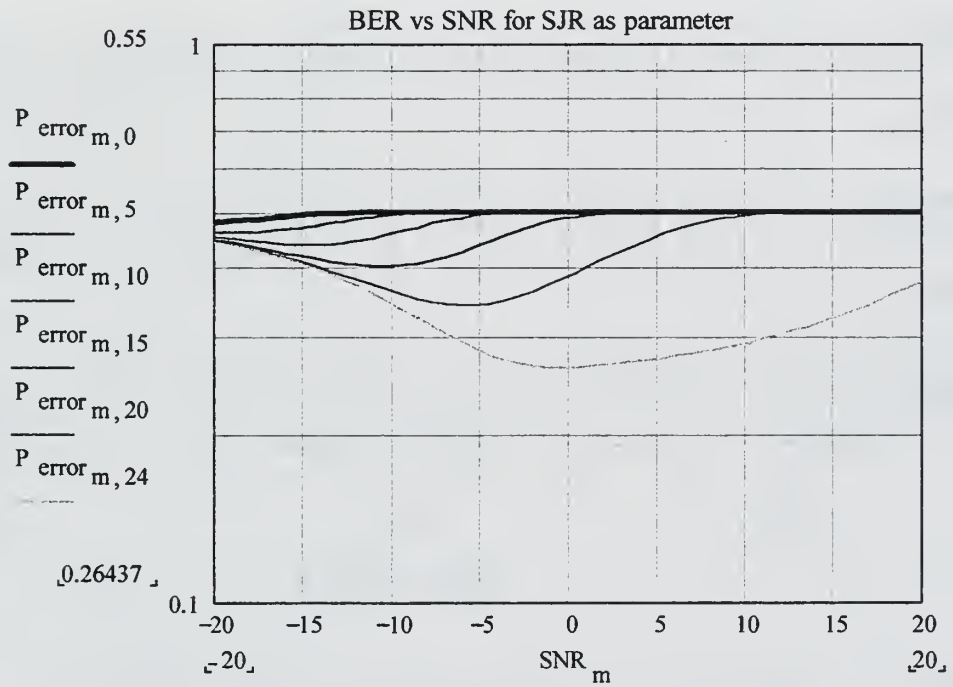

Figure 61. Probability of Bit Error versus SNR for Coherent BFSK with AWGN and Interference.

Six curves for SJR ratios from -20 dB to 0.4 dB in 4.1 dB increments are shown in Figure 61.

The theoretical BER for coherent BFSK as a function of the SJR is shown in Figure 62. Seven curves for SNR ratios from -20 dB to 4.5 dB in 4.1 dB increments are shown.

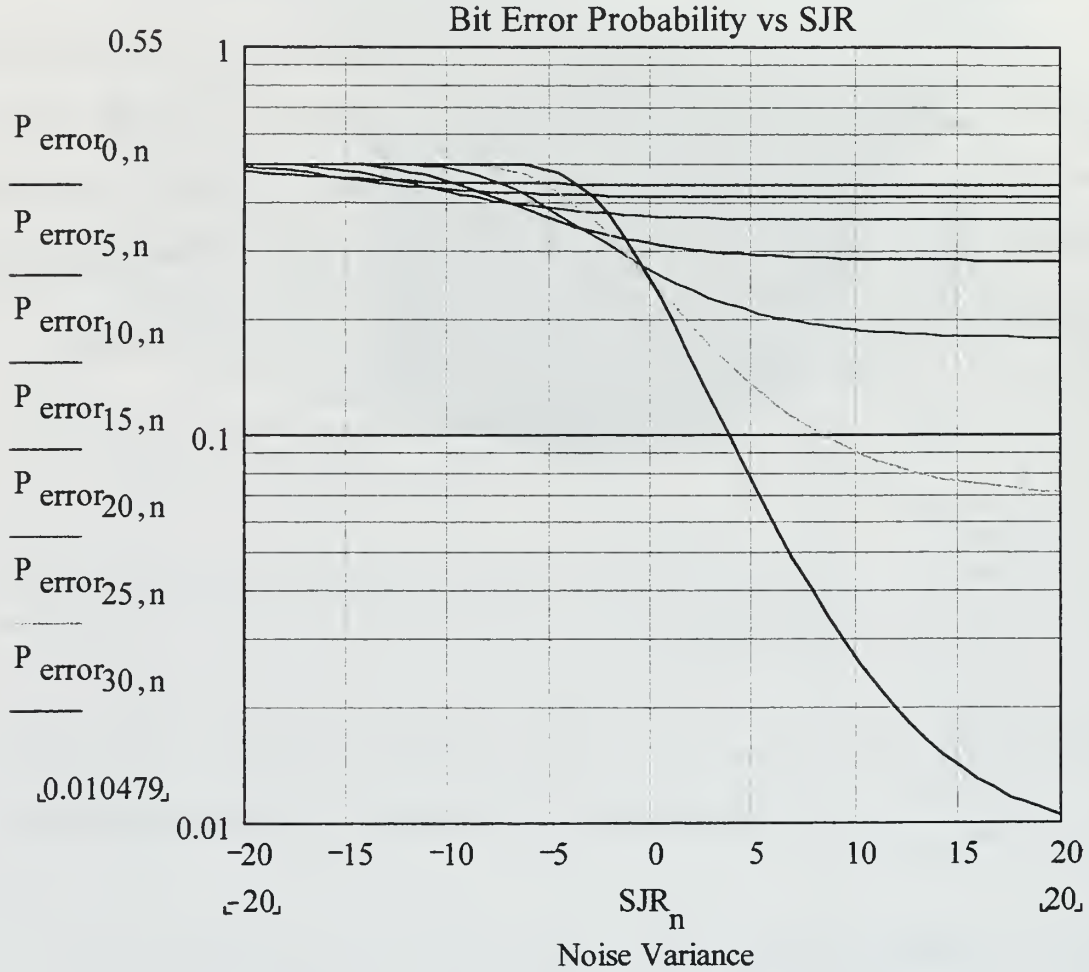

Figure 62. Probability of Bit Error versus SJR for Coherent BFSK with AWGN.

LIST OF REFERENCES.

1. W. Wang, *Communications Toolbox User's Guide*, The Math Works Inc.,2nd ed., 1996.

2. *Simulink User's Guide*, The Math Works Inc., 1993

3. Bernard Sklar, *Digital Communications*, Prentice Hall, New Jersey, 1988.

4. M. Jeruchim et al, *Communication System Simulation*, Plenum, 1992.

5. I. Glover and P. Grant, *Digital Communications*, Prentice Hall, 1998.

6. Floyd M. Gardner and John D. Baker, *Simulation Techniques*, John Wiley and Sons Inc., 1997.

7. Lebaric Jovan, *Course Notes for EC 4550 Digital Communications*, Naval Postgraduate School, Monterey, CA, 1998.

8. Lebaric Jovan, *Course Notes for EC 2500 Communications Systems*, Naval Postgraduate School, Monterey, CA, 1997.

9. The Matlab Expo, *An Introduction to Matlab, Simulink, and the Matlab Toolboxes*, Inc., Natick, MA, April 1993.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center.....2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library.....2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Chairman, Code EC.....1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

4. Chairman, Code IW.....1
   Information Warfare Academic Group
   Naval Postgraduate School
   Monterey, CA 93943-5121

5. Professor Clark Robertson, Code EC/Rc.....1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

6. Professor David Jenn, Code EC/Jn.....1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

7. Dr. Jovan Lebaric, Code EC/Le.....2
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

8. Embassy of Greece.....1
   Army Attache
   2228 Massachusetts Avenue, N.W.
   Washington, DC 20008

9. Konstantinos Tsiridis.....3
   Ierolohiton 41
   Drama 66100
   GREECE

66 553NPS TH 3792
11/99 22527-106 NIULE

DUDLEY KNOX LIBRARY
3 2768 00366450 9